



**STINFO COPY**  
**United States Air Force**  
**Research Laboratory**

**Cultural Modeling of Command  
and Control Echelons**

Rebecca Mui  
Tim Bagnall  
Nils LaVine  
Steven Peters  
Bob Sargent  
Riley Goodin  
Ramiro Ramos

Micro Analysis & Design, Inc.  
4949 Pearl East Circle, Suite 300  
Boulder, CO 80301-2477

April 2004

Final Report for the Period September 2001 to April 2004

20041025 002

Human Effectiveness Directorate  
Warfighter Interface Division  
Cognitive Systems Branch  
2698 G Street  
Wright-Patterson AFB OH 45433-7604

Approved for public release; distribution is unlimited.

**BEST AVAILABLE COPY**

## **NOTICES**

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service  
5285 Port Royal Road  
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center  
8725 John J. Kingman Road, Suite 0944  
Ft. Belvoir, Virginia 22060-6218

## **TECHNICAL REVIEW AND APPROVAL**

AFRL-HE-WP-TR-2004-0112

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public.

This technical report has been reviewed and is approved for publication.

## **FOR THE COMMANDER**

//Signed//

BRADFORD P. KENNEY, Lt Col, USAF  
Deputy Chief, Warfighter Interface Division  
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</p>			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	April 2004	Final - September 2001 - April 2004	
4. TITLE AND SUBTITLE		5. FUNDING NUMBERS	
Cultural Modeling of Command and Control Echelons		C: F33615-01-C-6076 PE: 63231F PR: 4923 TA: 04 WU: 06	
6. AUTHOR(S)		8. PERFORMING ORGANIZATION REPORT NUMBER	
Rebecca Mui, Tim Bagnall, Nils LaVine, Steven Peters, Bob Sargent, Riley Goodin, Ramiro Ramos			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	
Micro Analysis & Design, Inc. 4949 Pearl East Circle, Suite 300 Boulder, CO 80301-2477		Air Force Research Laboratory, Human Effectiveness Directorate Warfighter Interface Division Air Force Materiel Command Cognitive Systems Branch Wright-Patterson AFB OH 45433-7604	
11. SUPPLEMENTARY NOTES		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
		AFRL-HE-WP-TR-2004-0112	
12a. DISTRIBUTION AVAILABILITY STATEMENT		12b. DISTRIBUTION CODE	
Approved for public release; distribution is unlimited.			
13. ABSTRACT (Maximum 200 words)			
<p>The Air Force Research Laboratory (AFRL) funded Micro Analysis and Design (MA&amp;D) to address the needs of the Cultural Modeling of Command and Control (CMC2) project from September 2001 to April 2004. Specifically, the technical effort consisted of three primary objectives as follows: 1)Investigate cultural factors and add cultural modeling capabilities to an existing human performance modeling tool to allow users to easily inject cultural effects into a human performance model; 2)Create a client-server architecture between the Human Performance Model (HPM) tool and a constructive simulation using High Level Architecture (HLA) and Direct Shared Memory (DSM) to allow entities to receive higher fidelity behavioral representation from an external simulation tool; 3)Develop a model of an Integrated Air Defense System (IADS) for two different cultures to demonstrate the functionality of the enhanced HPM tool as well as the interaction of the HPM with a constructive simulation operating in a client-server environment. Two existing technologies were selected for this effort: the Combat Automation Requirements Testbed (CART) and the Joint Integrated Mission Model (JIMM). CART was chosen as the human performance modeling tool to be upgraded with cultural modeling capabilities and was modified to communicate using a client-server architecture via HLA and DSM. JIMM was selected as the constructive simulation to link with the enhanced CART HPM tool to enable it to receive higher fidelity human performance representations. The end product of the CMC2 effort is a toolkit capable of performing cultural modeling within an HPM tool and benefiting from the many advantages of a stand-alone HPM operating through a client-server architecture.</p>			
14. SUBJECT TERMS		15. NUMBER OF PAGES	
Cultural Modeling of Command and Control (CMC2) Combat Automation Requirements Testbed (CART) Joint Integrated Mission Model (JIMM) Opposing Forces (OPFOR) Intelligent Controller Node Integrated Air Defense System		94	
17. SECURITY CLASSIFICATION OF REPORT		18. SECURITY CLASSIFICATION OF THIS PAGE	
UNCLASSIFIED		UNCLASSIFIED	
19. SECURITY CLASSIFICATION OF ABSTRACT		20. LIMITATION OF ABSTRACT	
UNCLASSIFIED		UL	

THIS PAGE INTENTIONALLY LEFT BLANK

AFRL/WS  
-04-0624

## Table of Contents

<b>1.0</b>	<b>PROGRAM OVERVIEW .....</b>	<b>1</b>
1.1	BACKGROUND.....	1
1.2	CMC2 PROJECT OVERVIEW.....	2
<b>2.0</b>	<b>EXISTING TECHNOLOGIES .....</b>	<b>2</b>
2.1	COMBAT AUTOMATION REQUIREMENTS TESTBED (CART) .....	2
2.2	JOINT INTEGRATED MISSION MODEL (JIMM).....	3
2.3	CLIENT-SERVER ARCHITECTURE .....	4
<b>3.0</b>	<b>TECHNICAL DEVELOPMENT .....</b>	<b>6</b>
3.1	CULTURAL MODELING TOOL DEVELOPMENT.....	6
3.1.1	<i>Cultural Modeling</i> .....	6
3.1.2	<i>CMC2 Tool</i> .....	8
3.2	CART-JIMM CLIENT-SERVER DEVELOPMENT .....	13
3.2.1	<i>CART Middleware</i> .....	13
3.2.2	<i>CART-JIMM High Level Architecture (HLA)</i> .....	15
3.2.3	<i>CART-JIMM Direct Shared Memory (DSM)</i> .....	18
3.3	CART-JIMM CLIENT-SERVER PERFORMANCE .....	22
3.3.1	<i>Testing Overview</i> .....	22
3.3.2	<i>Small Scale Scenario Test Summary</i> .....	23
3.3.3	<i>Large Scale Scenario Test Summary</i> .....	24
3.3.4	<i>Performance Summary</i> .....	25
<b>4.0</b>	<b>IADS MODEL DEVELOPMENT .....</b>	<b>26</b>
4.1	INTEGRATE AIR DEFENSE SYSTEM (IADS).....	26
4.2	IADS SCENARIOS .....	27
4.2.1	<i>IADS scenario development requirements</i> .....	29
4.2.2	<i>Blue Force Representations</i> .....	30
4.2.3	<i>Red Force Representations</i> .....	30
4.3	CULTURAL FACTORS FOR IADS SCENARIO .....	31
4.4	IADS CULTURAL DEMONSTRATION RESULTS .....	32
<b>5.0</b>	<b>ACCOMPLISHMENTS .....</b>	<b>34</b>
5.1	FOLLOW-ON WORK .....	35
<b>6.0</b>	<b>ACKNOWLEDGEMENTS .....</b>	<b>35</b>
<b>7.0</b>	<b>REFERENCES .....</b>	<b>36</b>
<b>APPENDIX A: JOINT INTEGRATED MISSION MODEL (JIMM) DESIGN DOCUMENT FOR IMPLEMENTING TIME MANAGEMENT .....</b>		<b>37</b>
<b>APPENDIX B: INTERFACE CONTROL DOCUMENT FOR DATA INTERACTIONS BETWEEN JIMM AND THE CART MIDDLEWARE .....</b>		<b>56</b>
<b>APPENDIX C: CART HUMAN PERFORMANCE MODEL FOR IADS SCENARIO .....</b>		<b>64</b>

## List of Figures

FIGURE 2-1. CLIENT-SERVER ARCHITECTURE .....	5
FIGURE 2-2. CLIENT-SERVER SUBSCRIPTION PROCESS .....	6
FIGURE 3-1. CART TASK NETWORK .....	9
FIGURE 3-2. CART CULTURAL MODELING OPTION .....	10
FIGURE 3-3. CART EXECUTE OPERATIONS MODEL MENU OPTIONS .....	10
FIGURE 3-4. CULTURAL PARAMETER INTERFACE .....	11
FIGURE 3-5. CULTURAL MACRO INTERFACE .....	11
FIGURE 3-6. CART CULTURAL TEMPLATE MENU .....	12
FIGURE 3-7. CULTURAL TEMPLATE INTERFACE .....	13
FIGURE 3-8. CART MIDDLEWARE DESCRIPTION .....	14
FIGURE 3-9. JIMM-CART HLA ARCHITECTURE .....	16
FIGURE 3-10. CART-JIMM HLA FEDERATE CONFIGURATION .....	17
FIGURE 3-11. ALTERNATE CART-JIMM DIRECT SHARED MEMORY CONFIGURATION .....	19
FIGURE 3-12. CART-JIMM DIRECT SHARED MEMORY CONFIGURATION .....	19
FIGURE 3-13. DSM CONSTELLATION EXECUTIVE DATA MEMBERS .....	21
FIGURE 4-1. GENERIC IADS LAYOUT .....	26
FIGURE 4-2. JIMM-CART INTERFACE .....	27
FIGURE 4-3. JIMM PREWAR SCREENSHOT .....	28
FIGURE 4-4. JIMM TRADITIONAL SCENARIO SCREENSHOT .....	29

## List of Tables

TABLE 3-1. TEST SYSTEM CONFIGURATIONS .....	22
TABLE 3-2. SMALL-SCALE SCENARIO TEST SYSTEM CONFIGURATION .....	24
TABLE 3-3. SMALL-SCALE SCENARIO TEST RESULTS .....	24
TABLE 3-4. LARGE-SCALE SCENARIO TEST SYSTEM CONFIGURATION .....	25
TABLE 3-5. LARGE-SCALE SCENARIO TEST RESULTS .....	25
TABLE 4-1. IRAQ ASSIGNMENT SAMPLES FROM A TRADITIONAL SCENARIO .....	33
TABLE 4-2. NORTH KOREA ASSIGNMENT SAMPLES FROM A TRADITIONAL SCENARIO .....	33
TABLE 4-3. IRAQ ASSIGNMENT SAMPLES FROM A PREWAR SCENARIO .....	33
TABLE 4-4. NORTH KOREA ASSIGNMENT SAMPLES FROM A PREWAR SCENARIO .....	34
TABLE 4-5. IRAQ ASSIGNMENT SAMPLES FROM A SAMBUSH SCENARIO .....	34
TABLE 4-6. NORTH KOREA ASSIGNMENT SAMPLES FROM A SAMBUSH SCENARIO .....	34

## **1.0 Program Overview**

This report provides a summary of the work performed by Micro Analysis & Design (MA&D) and Northrop Grumman Mission Systems (NGMS), formerly TRW, on the Cultural Modeling of Command and Control (CMC2) project funded by the U. S. Air Force Research Laboratory (AFRL) under contract F33615-01-C-6076. This document is the final report that discusses activities performed from September 2001 to April 2004.

### **1.1 Background**

Computer Generated Forces (CGFs) and constructive simulations are continuing to play an increasingly important role in assistance to trainers, mission planners, modelers, system analysts and others interested in creating and using simulations for purposes such as assessment and training. Constructive simulations have been developed that model characteristics of vehicles, weapons and other equipment and have been verified and validated. However, in spite of these accurate models, most of these constructive simulations use relatively simplistic and predictable representations of behaviors of the humans operating and interacting with this equipment. This deficiency can easily discredit the final results and is widely accepted to be in dire need of improvement.

Much work has been done in the development of human behavior representation outside of CGF applications, primarily in the areas of cognitive processing, workload, training/experience, and environmental and other stressors. A recently recognized need is to include cultural biases or factors in the representation of human performance and decision making. Of particular interest to the military are the cultural or country biases of opposing forces. With a good understanding of the culture or country a commander is up against, he can be in a more informed position to take the more appropriate action to accomplish his goal whereas not taking these factors into account may result in an unanticipated and unwanted outcome. Recognition of the influence of cultural effects has driven a need to create a tool that will allow an analyst to more accurately represent human behavior including the impact of cultural factors on those behaviors.

Many standalone tools have been developed to refine the representation of human behavior. Integration of these advanced human performance modeling (HPM) capabilities with robust applications in the constructive simulation environment would provide analysts with powerful tools to enable them to create more accurate models. A typical approach to improvements in Human Behavioral Representation (HBR) within constructive simulations has been to embed HPMs directly within the CGF application. However, combining code in this manner increases the complexity of the constructive simulation software and decreases its execution speed. In addition, once the higher fidelity HPMs are embedded into a CGF, validating and maintaining those models becomes much more difficult and impractical. Without the ability to validate and maintain the HPM, their credibility is highly suspect and typically unaccepted.

A method that would provide an ideal environment for modification and validation of the HPM is to create a standalone, external application and link it to the constructive simulation through a client-server architecture. This architecture would make the use and upkeep of HPMs and CGFs much more practical. By including HPMs via a client-server architecture, changes and improvements could be made to the HPM without requiring any changes to the CGF software. Employing this architecture would remove the potential for introducing unanticipated and undesirable modifications to the CGF code base, and would no longer require a CGF software developer to delve into the morass of CGF code when dealing with HPMs. The standalone HPM would provide HBR to constructive simulations through an external, common architecture. Any

and all CGFs that support this external architecture could then easily use the services of the HPM. Decoupling the HPM from the CGF has the added benefit of giving the CGF the ability to easily incorporate numerous HPMs for high fidelity modeling of human behaviors without sacrificing processing capability or compromising validated CGF code.

This CMC2 effort recognizes the two needs expressed here – a capability to perform cultural modeling within an HPM tool and benefit from the many advantages of a standalone HPM operating through a client-server architecture.

## 1.2 CMC2 Project Overview

This technical effort consisted of three primary objectives as follows:

- 1) Investigate cultural factors and add cultural modeling capabilities to an existing human performance modeling tool to allow users to easily inject cultural effects into a human performance model.
- 2) Create a client-server architecture between the HPM tool and a constructive simulation using High Level Architecture (HLA) and Direct Shared Memory (DSM) to allow entities to receive higher fidelity behavioral representation from an external simulation tool.
- 3) Develop a model of an Integrated Air Defense System (IADS) for two different cultures to demonstrate the functionality of the enhanced HPM tool as well as the interaction of the HPM with a constructive simulation operating in a client-server environment.

Two existing technologies were selected for this effort: the Combat Automation Requirements Testbed (CART) and the Joint Integrated Mission Model (JIMM). CART was chosen as the human performance modeling tool to be upgraded with cultural modeling capabilities and was modified to communicate using a client-server architecture via HLA and DSM. JIMM was selected as the constructive simulation to link with the enhanced CART HPM tool to enable it to receive higher fidelity human performance representations.

In the following sections, CART and JIMM are described in further detail and the changes to those applications resulting from this effort are discussed. The results of the cultural modeling research and the IADS model and scenario development are also presented.

## 2.0 Existing Technologies

### 2.1 Combat Automation Requirements Testbed (CART)

CART is the result of an AFRL effort to create a modeling and simulation interface that allows human performance and behavior considerations to be included in weapon system design and acquisition. CART facilitates the inclusion of crew system performance and behavior considerations early in complex system development. CART is an extension of the Army Research Laboratory's (ARL) Improved Performance Research Integration Tool (IMPRINT) with the addition of goal orientation modeling capability and the addition of an adaptive simulation interoperability environment. This environment allows CART models to communicate with other simulations through a HLA Run Time Infrastructure (RTI) and with first principle models and applications directly through Component Object Model (COM) Services.

The goal-oriented behavior modeling capability of CART allows the user to anticipate possible mission interruptions or additional workload that might occur during a specific scenario. Users are able to represent complex, competing performance requirements for a system's mission, and evaluate a system based on how those requirements affect the overall mission performance.

CART simulation models can act as a federate within an HLA-compliant federation using a CART Middleware that allows a user to send data, object attributes and interactions across the federation. CART HPMs run in a time managed mode with other simulations, and interactions with other simulations may affect the way that CART models run. CART models can also interact directly with first principle models and applications and allows high fidelity first principle model considerations to be included dynamically in CART HPMs.

The capabilities and flexibility of CART make it the ideal starting point for creating an HPM tool with cultural modeling capabilities that could provide services to a CGF using a client-server architecture.

## 2.2 Joint Integrated Mission Model (JIMM)

JIMM is a general purpose, data-driven, conflict simulation/environment generator. The descriptions in the previous sentence portray the enormous aspects of JIMM and its capabilities.

*General purpose* – refers to the capability in JIMM to describe conflicts across a broad range of issues, detail, geographical extent, and time. JIMM has a balanced approach to all aspects of the conflict. JIMM internally knows nothing, nor does it need to know anything, about military, political, or other issues that may be of importance to the decision maker. It is up to the user to specify these issues to the level of detail necessary.

*Data-driven* – refers to the fact that each entity in the scenario is specified through parameters, interactions, and interrelationships defined by the JIMM databases. JIMM's simulation engine performs three basic functions: 1) changes the state of a player object or environment; 2) generates events; and, 3) exports/imports data to or from external assets. JIMM players are based on six generic processes: moving, thinking, sensing, communicating, shooting, and disrupting or jamming.. JIMM is a generic modeling environment without hard-coded entities of any kind. Every entity is a combination of systems, tactics and susceptibilities represented by these generic processes.

*Conflict* - refers to situations in which there is contention over the use or control of resources. The contention could be within someone's mind, or it could be a widespread conflict among several factions. The nature of the conflict can change over time. The conflict might, for example, start as a relatively peaceful discussion between the participants and could escalate to a highly energized, lethal dispute. JIMM contains very few internal assumptions about the nature of the conflict.

*A simulation* - represents the changes in the conflict over time. A simulation is dynamic, not static. Thus, the user needs to set up the initial conditions for the simulation and the basic assumptions or rules in JIMM about cause and effect that drive the changes in the conflict as time progresses.

*Environment Generator* - refers to the fact that JIMM performs an important function that is missing from many other simulations. In fact, the environment generation aspect contributes to the "completeness" of JIMM. Environment generation refers to the need (and ability) to provide realistic backdrops for other computer simulations, human-in-the-loop simulators, or hardware-in-the-loop stimulators. This feature allows the user to take advantage of higher fidelity resources that otherwise would not be utilized in an integrated representation of a conflict. JIMM has been used to represent a full spectrum of simulated entities or players in distributed networks with other simulations, simulators, hardware, and human-in-the-loop operators. JIMM has also been used to represent aspects of the simulated environment not represented by other components in a network or model federation. Examples include manned flight simulators or human-in-the-loop decision makers operating in a virtual environment provided by JIMM and missile simulators

integrated with JIMM providing higher fidelity representation of Surface-to-Air missile (SAM) units. While these examples have executed in real time, constructive analyses have also been conducted using JIMM integrated with other computer simulations at faster than real time.

In addition to the simpler functions of movement, sensing, terrain occulting, and weapons firing, JIMM incorporates concepts such as perception, command and control, communication, jamming, reactive logic, planning, and the creation and absorption of players and resources. These concepts will be expanded upon and finely tuned to meet requirements of specific CART modeled entities in a combined simulation through the CART-JIMM architecture.

JIMM is also a highly flexible, real-time threat generator and run time executive that supports exercises containing constructive, virtual, and live players. JIMM permits the substitution of any constructive system, platform, or player by its virtual or live counterpart. CART models will be used to replace human thinkers in the JIMM virtual environment to provide high fidelity, human performance modeling.

JIMM is capable of executing in both a stand-alone mode and a networked mode of operation. JIMM is DIS and HLA compliant and supports DSM, and, therefore, can link to a variety of simulators, stimulators and simulations as mentioned above. The CART-JIMM architecture will utilize both HLA and DSM. Figure 2-1 shows a high level depiction of the JIMM architecture and capability.

JIMM is an event-driven simulation in which all events are ordered through a single event queue. When JIMM executes within a federation, each event in the queue must be released at the appropriate time to maintain proper synchronization with the other federates. A challenge of incorporating the client-server architecture and the “middleware” of CART into the architecture of JIMM was to maintain maximum federation execution speed while also keeping CART HPMs in synch with JIMM. Both HLA Time Management services and a DSM time management scheme were employed to accomplish these goals.

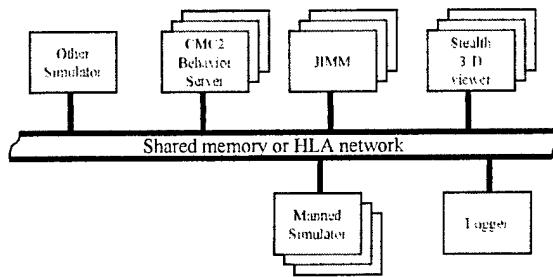
Modifications to JIMM were necessary to create the client-server architecture. This change involved defining and implementing a common communication and data transfer protocol. Time management services were also added to provide the time stepped synchronization needed between CART and JIMM to achieve higher fidelity through run-time interactions.

### **2.3 Client-Server Architecture**

CGF programs have traditionally implemented improvements by incorporating HPM behavioral representation directly into the code base. This method has significantly contributed to a continually growing software code base that has become an increasingly complex and correspondingly difficult to maintain set of simulation tools. If, instead, the HPMs could function external to the CGF tools, the development, validation and maintenance of the HPMs become much simpler and the capabilities of the CGF would not be compromised.

One method of enabling high fidelity HPMs that reside in an external application to interact with CGF entities is through a client-server architecture. In a client-server architecture, the “client” makes requests for and is provided with services from the “server.” As depicted in Figure 2-1, CMC2 Behavior Servers reside in the federate and interact with constructive simulations, such as JIMM, using a client-server architecture implemented over a common network. The HPMs reside in the behavior servers and will provide the higher fidelity human behavior representations to the CGFs. Using this architecture, the behavioral representation is off loaded from the CGF system (the client) to an external behavioral server that can facilitate the inclusion of variable fidelity entity behaviors as well as much more complex entity behaviors than is available within the CGF system itself. Additionally, the architecture between the client and the server only

requires a common communication method and a common protocol. Once in place, any CGF client could request services from any HPM residing in a behavior server.



**Figure 2-1. Client-server architecture.**

The client-server architecture also allows the CGF user to select the type and number of entities within the CGF for which a higher fidelity human behavior representation is desired. In this manner, when a HPM is desired, the CGF is not required to perform HPM processing for every entity defined within it - only those entities that are selected for subscription to an external behavior server. By not increasing the complexity of the CGF code and simultaneously providing HPM processing only where needed, the CGF will perform efficiently.

In a client-server architecture, a subscription process is used to coordinate connections between the entities desiring HBRs and the HPM services that they will require. When the simulation starts, the CGF client sends a request for subscription to a server via data interactions, as shown in Figure 2-2. Behavior servers that support that type of service request will respond to the request with a data interaction specifying that it can accept a new entity. The CGF then selects one of the responding servers and completes the subscription process by acknowledging the reply with another data interaction to the selected server. The behavior server, in turn, confirms the subscription with another data interaction. This subscription process, developed by MA&D, allows for multiple servers and includes load balancing [1].

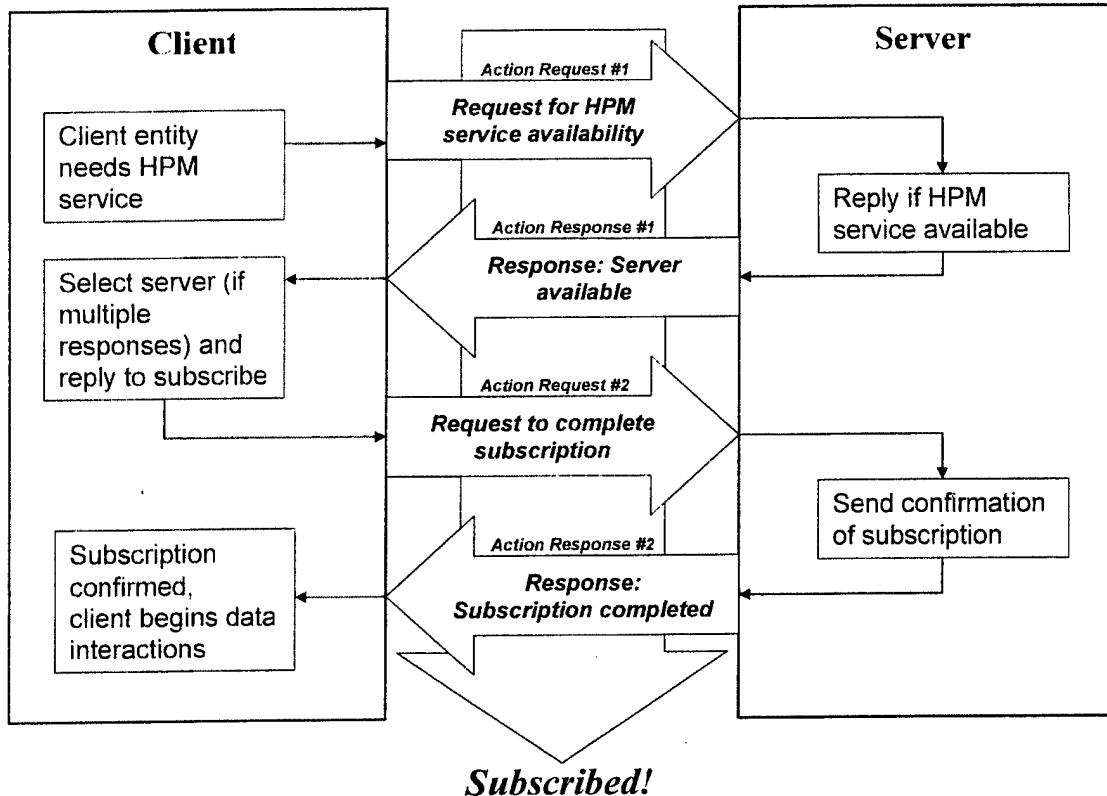


Figure 2-2. Client-server subscription process.

This client-server architecture has been successfully implemented on several projects to provide external HBRs to constructive simulations including Modular Semi-Automated Forces (ModSAF), Dismounted Infantry Semi-Automated Forces (DISAF), OneSAF Testbed Baseline (OTB), and Joint Semi-Automated Forces (JointSAF). Applying the client-server approach to the CMC2 project enables cultural effects to be modeled using an external HPM tool and then added to a CGF system without embedding those effects directly into the CGF.

### 3.0 Technical Development

#### 3.1 Cultural Modeling Tool Development

##### 3.1.1 Cultural Modeling

Models of human performance capabilities in CGFs such as JIMM tend to be in terms of quantifiable factors such as workload and environmental stressors. Advancements in the capabilities of HBR tools have led to the ability to represent more complex behaviors, particularly in cognitive tasks such as decision-making. More recently, analysts and the user community have become increasingly interested in the role that culture plays in decision making and other tasks.

For this project, cultural modeling was defined as *the application of cultural or country based influences on human behavior or performance within a human performance model*. Given a situation with the same physical conditions and the same resources, people from different cultures or countries may react to the situation and apply their resources differently. A cultural difference exists when the “average” reaction of a population from one culture differs from that of another.

It is important to note that cultural effects are biases within a population to behave in one manner over another and care must be taken in the definition of such biases and the interpretation of the results.

The first task in creating a cultural modeling tool was to gather data on cultural factors and the differences exhibited by various cultures. Volumes of literature can be found which discuss all the different aspects of cultures from the descriptions of behaviors, values, customs and attitudes that are universal across cultures to detailed studies that are specific to a single culture. Furthermore, subcultures exist within cultures, making the study of cultural differences even more complex and obtaining data more elusive.

Since one of the objectives of the project was to apply the effects of cultural factors to an IADS model, the cultural research began with an investigation into human performance within IADS systems. IADS subject matter experts (SMEs) were interviewed to obtain information related to IADS operations and any perceived differences in performance by different cultures. While the SMEs were extremely knowledgeable on the equipment and operational requirements within an IADS, they did not have any data on human performance. A literature search on IADS operations also provided no available relevant data.

The cultural research was then expanded to explore cultural influences in military operations. Some studies noted the importance of recognizing cultural differences in human performance but no studies provided quantitative data. However, one study provided an excellent summary of the sources of cultural differences and methods for studying and classifying these differences. Klein et al. developed the Cultural Lens model to help leaders understand the differences between cultures and the effects of these differences on multinational collaboration in military C2 operations [2].

The Cultural Lens is a concept for a tool that would help a leader from one culture view a situation from the perspectives of other cultures. Insight into the perspectives of other cultures would give the leader a better understanding of why or how the leader of another culture may react differently to a situation. Through the Cultural Lens, a leader may also see that his own actions could create unintended reactions from another culture. A Cultural Lens would most likely increase the effectiveness of any multinational collaborative operation.

Klein et al., references a comprehensive study by Hofstede [3] on the cultural differences found within a large multinational corporation. Hofstede defines four cultural dimensions that he uses to classify the differences between cultures. These dimensions, while not directly applicable to C2 or IADS operations, provided insight into developing more relevant classifications. Klein et al. derives two of their five cultural dimensions from Hofstede's dimensions of Power Distance and Uncertainty Avoidance. Taking these concepts one step further to focus on IADS, the cultural factors of Distribution of Power and Willingness to Take Risk were selected for the IADS scenario. An additional factor that could potentially influence IADS operations, Familiarity with the Enemy, was also selected for the IADS model. These factors are discussed in further detail in Section 4.3, Cultural Factor for IADS Scenario.

Despite the lack of quantitative cultural data, a taxonomy of cultural factors, such as those investigated by Hofstede and Klein, could be produced. However, any taxonomy would be highly dependent on the problem to be modeled, the relevance of the cultural factors included in the model and the data available to support those cultural factors. Typically, HPMs are designed to represent the human behavior for specific types of effects. For example, an HPM that calculates the effects of workload would contain an algorithm embedded into its code to determine workload levels. Rather than building a tool in this fashion, which would incorporate some predetermined taxonomy of cultural factors and their effects, a more useful tool would be

one that is more generic in nature and allows the user to compose any cultural factor that is relevant to the problem.

The cultural factors selected for the IADS are notional and could prove to have less impact than some other cultural factors. As data is found to support or disprove the effects of those cultural factors, new factors could be added to the model and the less relevant ones removed. A generic cultural modeling tool would be adaptable to this type of problem. The user could define and redefine cultural factors as necessary to incorporate the effects within the model that accurately reflect the supporting data.

For the CMC2 project, this capability of creating composable cultural factors was added to CART. Notional cultural factors as well as notional data were used to demonstrate the flexibility and functionality of the tool. Although no actual data was available to validate our selection of cultural factors, if this data became available, the HPM could easily be modified to incorporate new cultural data and/or cultural factors.

### **3.1.2 *CMC2 Tool***

#### **3.1.2.1 *CMC2 Tool Features***

The CMC2 effort addresses the issue of improving HPM capabilities for cultural modeling by enhancing CART to create the CMC2 tool. This tool was designed to function external to CGFs and provide human behavior representation using a client-server architecture. The CMC2 tool can be used by any CGF that has the capability to interact with the tool through the common client-server architecture. The CMC2 tool was used to incorporate notional cultural factors into the IADS demonstration model developed in JIMM, as discussed in Section 4, IADS Model Development.

The primary objective of the CMC2 tool was to give users the capability to represent various cultural factors within a model without the need to create new models for each culture. The CMC2 tool was designed with the following features:

- Provide the capability to compose culturally based parameters and macros for the model
- Allow users to create and save profiles, or templates, which define the relevant cultural parameters for each culture modeled and the values derived from cultural data that are assigned to those parameters
- Contain a library of parameter and macro definitions that can be applied to different models
- Allow users to easily modify parameters and templates to perform analyses on different scenarios with a single or multiple cultures

Users develop models within CART by first creating human performance task networks, breaking down actions into functions (or sub-networks) and tasks, as shown in Figure 3-1. The task network links these functions and tasks together and shows the overall process flow.

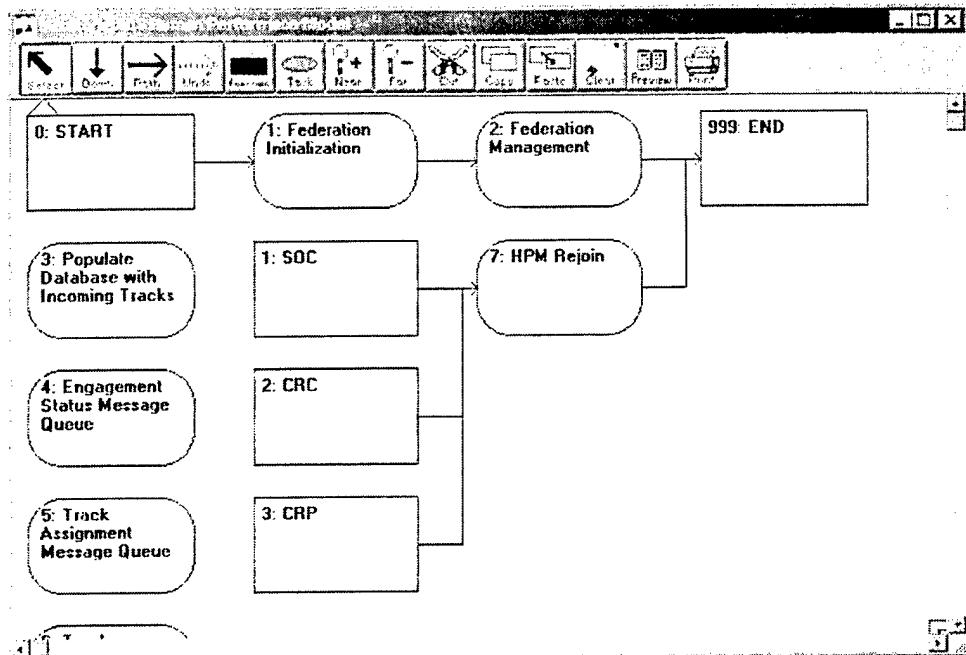


Figure 3-1. CART task network.

Once the network and tasks are created, the user defines the parameters and macros that will be used to describe the behaviors and processes carried out within each task. Parameters are user-defined variables that store values used by the model. Macros are functions, or series of steps, to be executed within the model. Utilizing the parameters and macros, the user defines behaviors and/or decisions that influence the performance of events within the tasks and the transitions between tasks within the task network. The parameters and macros are also used in release conditions, effects, task times, and decision branches. In addition, the user can assign performance related parameters such as task execution times, workload and other criteria that affect the execution of the model. A great deal of human performance and behavior complexity can be built into even relatively simple looking task network models.

A new feature, Cultural Modeling, was added to the CART tool, as shown in the CART user interface in Figure 3-2. After selecting Cultural Modeling from the menu, the user chooses from three options: Parameters, Templates or Macros. In the process of building a task network model in CART, the user defines model variables, or parameters, and macros. The Parameters and Macros options under the Cultural Modeling option are essentially the same as the standard CART variables and macros, respectively; however, the parameters and macros created under the Cultural Modeling option are marked to indicate that they are culturally based. By using the Parameters and Macros menu, the cultural parameters and macros are kept separate from the standard variables and macros. This feature allows the user to maintain good visibility of the parameters and macros that are based on cultural factors. Also, the cultural parameters are the only variables that can be used in the templates, described below.

In the Template option, the user creates cultural profiles by defining the values to be used for the cultural parameters in the model. One or more templates can be defined by the user. At execution time, the user selects the template that is to be used from the Execute menu, shown in Figure 3-3.

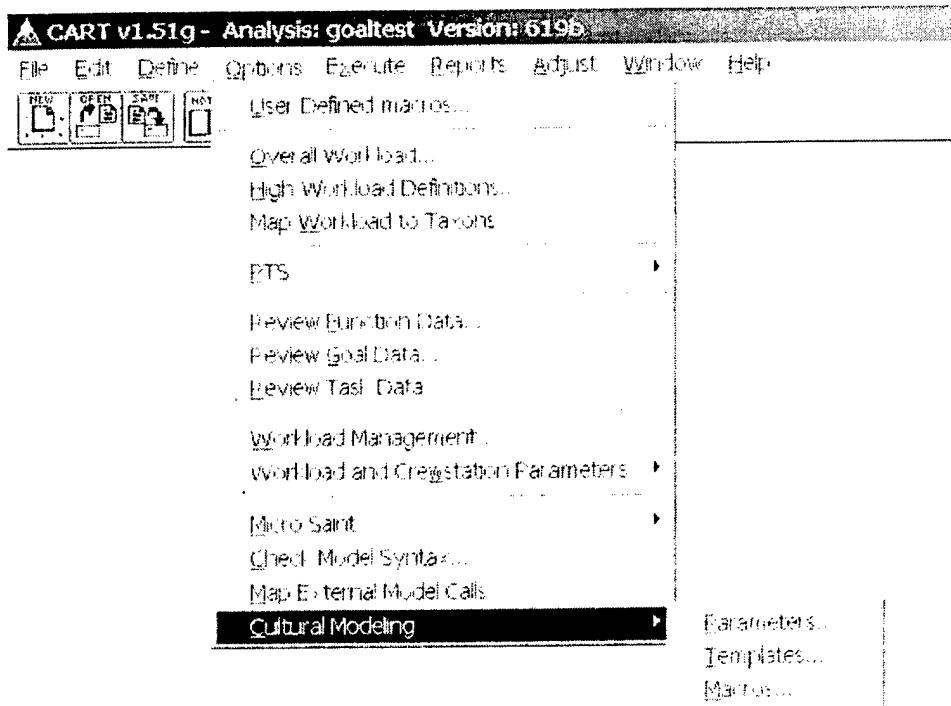


Figure 3-2. CART Cultural Modeling option.

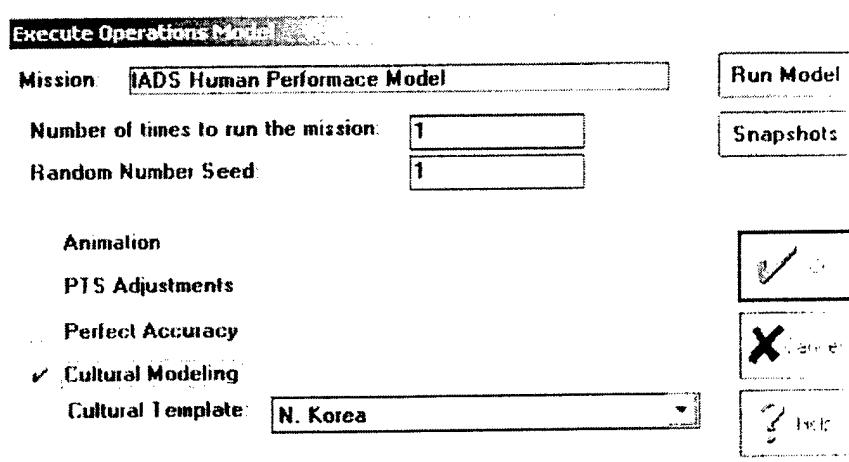


Figure 3-3. CART Execute Operations Model menu options.

### 3.1.2.2 Adding Cultural Parameters to a CART model

The Cultural Modeling options are used after creating the basic task network model. The user performs three steps to add the cultural parameters to the model:

- 1) *Define cultural parameters.* The user first defines the cultural parameters, or variables, which may vary between cultures (Figure 3-4). Defining these parameters through the cultural parameter interface provides easy access and visibility to the parameters used within the model to generate cultural effects.

The screenshot shows a software interface for managing cultural parameters. On the left is a table with columns: Variable Name, Type, External, and Description. The table contains the following data:

	Variable Name	Type	External	Description
72	eCultPrm_DP	Array of Integers	True	Distribution of
219	eCultPrm_FE	Array of Integers	True	Familiarity with
71	eCultPrm_WR	Array of Integers	True	Willingness to
222	proximityHi	Real	False	Highest rating
73	proximityLo	Real	False	Lowest rating
223	proximityMed	Real	False	Medium rating

On the right is a vertical toolbar with buttons: Open Array, Add..., Duplicate..., Delete, a pencil icon labeled 'Ok', a cancel icon labeled 'Cancel', and a help icon labeled 'Help'.

Figure 3-4. Cultural Parameter interface.

2) *Define cultural macros.* Next, the user defines the cultural macros for the model (Figure 3-5). The cultural macros contain the functions that may vary between cultures and utilize cultural parameters as well as other model variables in their definition. Like the parameters, these macros are created and contained in the cultural macro interface.

The screenshot shows a software interface for defining cultural macros. At the top, there are fields for 'Mission Name' (IADS Human Performance Model) and 'Macro Name' (cmPROXSCORE). Below these is a large text area containing the macro definition:

```
(CALCULATES THE PARTIAL THREAT SCORE FOR
PROXIMITY TO CRC)

(Calculates the distance from track to detecting CRC)
DISTANCE;

if (distFrmCrc <= proximityMin) then
  proxScore:= proxWeigh * proximityHi;

if (distFrmCrc > proximityMin & distFrmCrc <=
proximityMed) then proxScore:= proxWeigh *
proximityMed;

if (cultureA & distFrmCrc > proximityMin & distFrmCrc <=
proximityMed) then proxScore:= proxWeigh * proximityHi;
```

On the right is a vertical toolbar with buttons: Variable Catalog, a pencil icon labeled 'Ok', a cancel icon labeled 'Cancel', and a help icon labeled 'Help'.

Figure 3-5. Cultural Macro interface.

3) *Define cultural templates.* After defining the cultural parameters and macros, the user builds cultural profiles, or templates, in the cultural template interface shown in Figure 3-6. The user selects the cultural parameters that are affected by a selected culture and assigns the appropriate values for those parameters based on the cultural data (Figure 3-7). The CMC2 tool also allows the user to specify default values for each cultural parameter. The default value is used when cultural data for a parameter is not available.

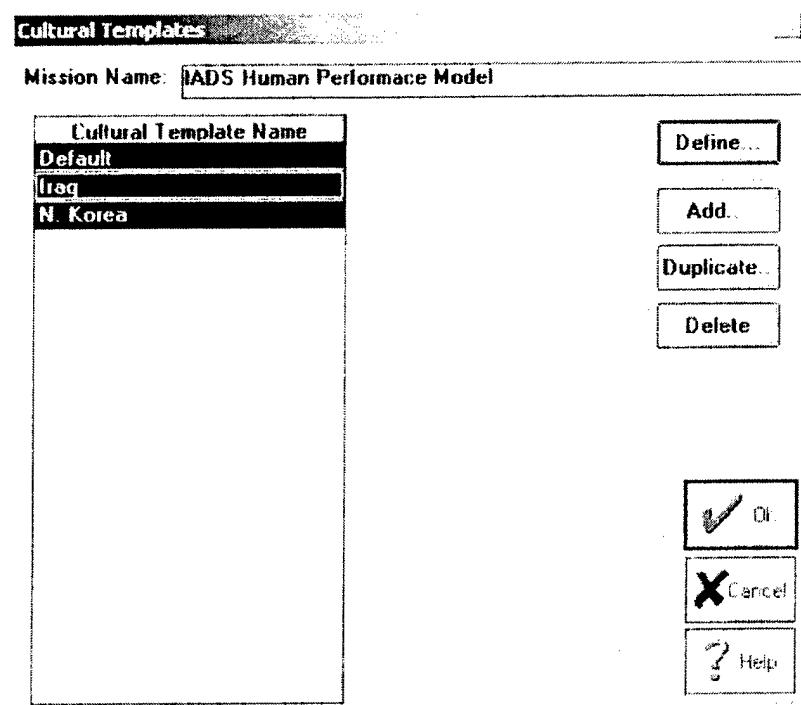


Figure 3-6. CART Cultural Template menu.

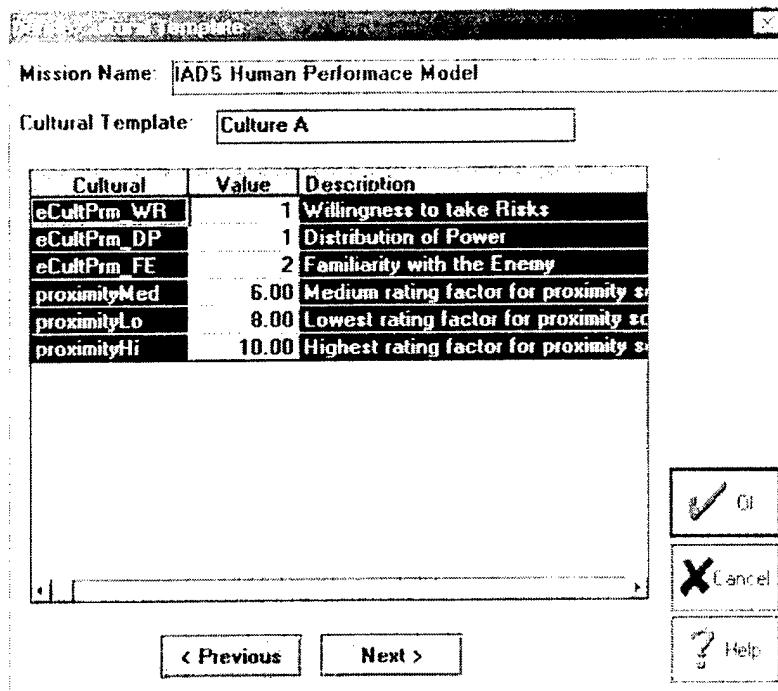


Figure 3-7. Cultural Template interface.

The user adds cultural effects to a CART model by incorporating the cultural parameters and macros within the task network. The cultural factors can affect not only the events that take place within a task but also the decisions and methods that link the tasks.

At run time, the user simply uses the cultural template interface to select templates for different analyses, as shown earlier in Figure 3-3, without the need to modify the baseline model. As more data is gathered to support the behavioral effects for additional cultures, the user can easily create a new template and add the new parameter values for that culture in the template interface.

The CMC2 project was focused on adding the capability to easily model *cultural* effects to the CART HPM tool. However, this method of adding composable parameters could also be extended to include other types of factors that would result in more complete profiles for simulating variations in behavior within the same model.

### 3.2 CART-JIMM Client-Server Development

#### 3.2.1 CART Middleware

The CART Middleware provides the external interface between the CART HPM tool and other applications. In previous efforts, MA&D designed and built the CART Middleware to provide single entity modeling to other constructive simulations such as ModSAF and the Fighter Requirements Evaluation Demonstrator (FRED).

For this CMC2 project, MA&D added the following upgrades to the CART Middleware to support the architecture between CART and JIMM:

- Multiple entity subscription capability via HLA
- A DSM message passing protocol
- CART/HLA style messages using the DSM message passing protocol

- Simple time management protocol for DSM

### 3.2.1.1 Middleware Overview

CART simulation models can serve as a federate within an HLA or a DSM compliant federation. To enable communication with other federates, MA&D developed two variations of a “middleware” application, one for HLA and one for DSM, for the CART HPM Environment (CHE). This middleware allows CART to communicate with any other federate that is HLA or DSM compliant.

The difference between the HLA middleware and DSM middleware lies within the method used by the middleware to interoperate with the federation executive. Figure 3-8 shows a general depiction of the CART Middleware. The shaded area in the figure marked “Federation Executive Module” represents the portion of the middleware that differs between HLA and DSM. For HLA federations, the HLA Link module enables communication for HLA networks. Likewise, the DSM Link module enables communication for DSM networks.

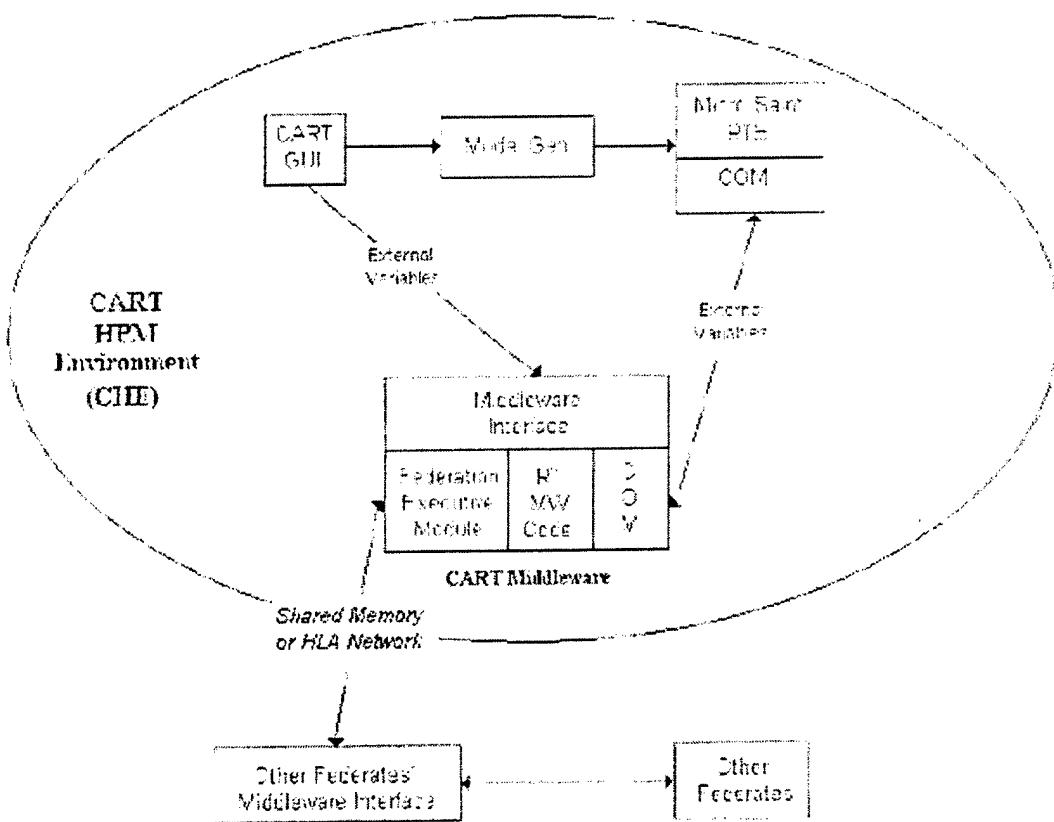


Figure 3-8. CART Middleware Description.

Prior to run-time, federation developers must define external variables – variables that are globally agreed upon data types for all federates - in the CART Graphic User Interface (GUI). In Figure 3-8, the arrow labeled “External Variables,” extending from the CART GUI to the Middleware Interface, represents the definition of these external variables. During federation

run-time, the Micro Saint Run Time Engine (RTE) uses Component Object Model (COM) services to communicate these external variables to the Middleware, as shown by the double-headed arrow connecting the two COM rectangles in Figure 3-8. Using either the “HLA Link” or “DSM Link”, the Run-Time Middleware Code and the Federation Executive Module communicates these external variables with other federates providing the final connection with the HLA or DSM network, respectively.

### **3.2.1.2 Developing the Middleware**

Prior to this effort, MA&D developed a version of the CART Middleware for the CHE on an HLA network. Improvements to this earlier version of the middleware were required to provide the functionality sought by the CART and JIMM federation. The earlier version offered only single entity modeling. That is, a CART application executing on one machine could provide high fidelity human performance modeling for only one entity within a federation. To accommodate other entities with high fidelity human performance modeling, additional federates running CART had to be added to the federation – one for each entity. Thus, as more entities needed high fidelity human performance modeling, the federation had to grow in size accordingly.

For this CMC2 project, a multiple entity Middleware was required. The improved middleware would allow one application of CART, as a single federate, to provide high fidelity human performance modeling to multiple entities. The efficiency gained by using one application of CART to provide all human performance modeling is having fewer federates. Nevertheless, if a complex federation required more processing power, additional instances of CART and the Middleware can execute on additional machines as supplementary federates, each accommodating one or more entities.

In addition to the enhancements made to support multiple entities, implementing DSM required further modification to the Middleware. MA&D took a modular design approach for implementing DSM by separating the functionality of the middleware unique to the HLA and DSM implementations into different modules, the HLA Link and DSM Link modules, respectively. Figure 3-8 represents these modules with the grayed box “Federation Executive Module.” The HLA specific code developed in the earlier versions of the middleware were used in the HLA Link module while new code developed to enable the middleware to communicate with other federates using the DSM protocol was used in the DSM Link module. The DSM Link module development involved coding a shared memory compliant, message passing protocol to allow external variables to communicate across the DSM network.

## **3.2.2 *CART-JIMM High Level Architecture (HLA)***

### **3.2.2.1 CART-JIMM HLA Integration**

One objective of the CMC2 project was to develop a client-server architecture between CART and JIMM using HLA (Figure 3-9). CART provides subscribed entities in JIMM with human performance models that are either not modeled at all or modeled with low (i.e., insufficient) fidelity within JIMM.

The HLA implementation of CART uses the Real-time Platform Reference Federation Object Model (RPR FOM) for communication with other federates. The RPR FOM contains Simulation Management (SIMAN) interactions that enable CART to send a custom defined data set across the federation that is not specified by other RPR FOM object attributes or interactions. The SIMAN interactions allow data flexibility by providing an open channel that permits any data in any format to be passed within an HLA federation simulation. JIMM and CART use these

SIMAN interactions in the client-server subscription process and for data interactions during simulation.

HLA Time Management services are also included to allow for temporal causation within the federations, assuring all events occur in chronological order.

When a JIMM entity requires the services of an external HPM, JIMM will make a request via HLA for subscription to a supporting external HPM. Multiple JIMM entities can be “served” by the same HPM and a single JIMM entity can be “served” by multiple HPMs.

By using this subscription process and data interactions within HLA, many other federated simulations can also benefit from the same high fidelity HPMs. Similarly, simply using the Middleware and following the same protocols, others could easily use alternative simulation tools to provide HBRs.

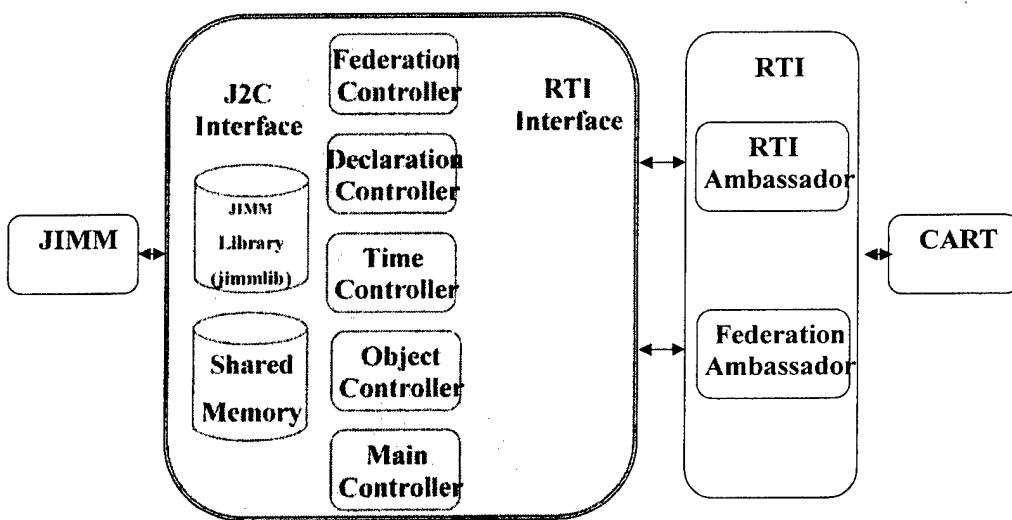


Figure 3-9. JIMM-CART HLA architecture.

### 3.2.2.2 Modifications to the CART Middleware for HLA

To enable the CART-JIMM architecture via HLA, the capability to handle multiple entity subscriptions was added to the CART Middleware. No other software modifications were required for the Middleware.

The communication protocol between CART and JIMM was established by defining data interactions. As noted earlier, the interactions take the form of SIMAN interactions within the HLA federation. Two subscription interactions and five possible data interactions can be sent between CART and JIMM:

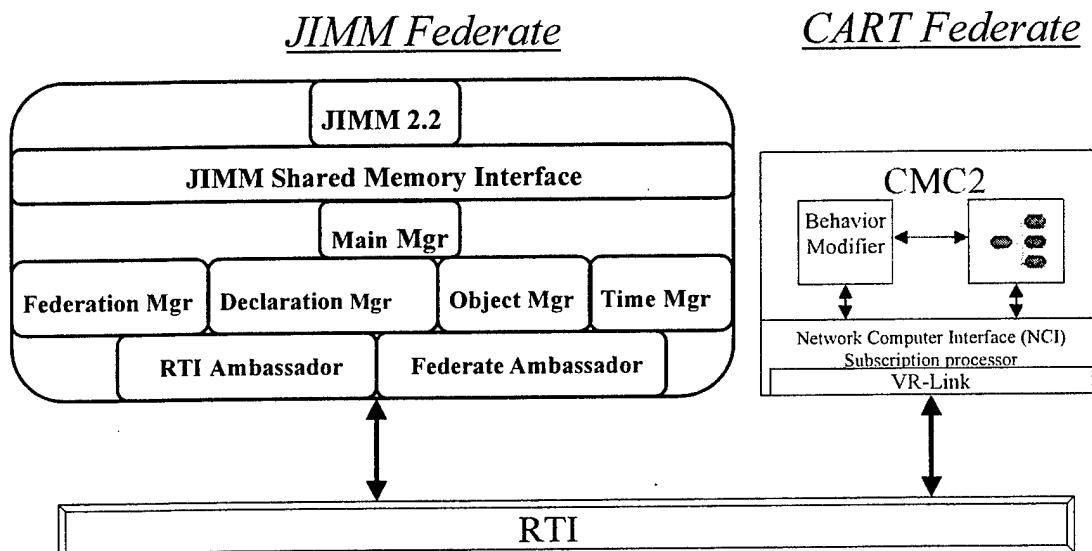
1. Subscription Services Data – data used to subscribe entities to specific HPM services
2. Initialization Data – initialization data for each entity subscribing for HPM service
3. Track Data – track information to be communicated between entities (Note: A track is an IADS term referencing an aircraft or other aircraft detected and known by the system’s radar)
4. Track Assignment – track assignment information
5. Cancel Track Assignment – track cancel information

6. HF Request – used by an entity to request track height information
7. Engagement Status – track engagement status information

These data interactions are defined in further detail in Appendix B - Interface Control Document for Data Interactions between JIMM and the CART Middleware.

### 3.2.2.3 Modifications to JIMM for HLA

JIMM and CART entities communicate using HLA through a series of data interactions as shown in Figure 3-10. The HLA RTI manages the delivery of these data interaction messages. To ensure that events occur in the proper sequence and at the appropriate times, HLA time management services were required for the CART-JIMM client-server federation. The CART-JIMM Federation is strictly time managed, meaning all federates are time regulated and time constrained. The RTI Time Management (TM) services ensure that the exchange of data interactions between federates occurs in a causally correct order.



**Figure 3-10. CART-JIMM HLA federate configuration.**

The following changes were made to JIMM to enable time management in HLA:

- The JIMM Federate design incorporates two simulation assets: the JIMM Asset itself and the HLA Interface Asset. In this effort, the JIMM Asset was modified to invoke Next Event Requests (NER) to the Runtime Infrastructure (RTI) and be called back with Time Advance Grants (TAG). Each asset stores time management variables in a common area of shared memory.
- New data structures were added to capture time management status.
- New Asset Action Dispatch messages were added to inform an asset of when Time Management procedures are in effect. These dispatch messages are part of the Multi-port Memory (MPM) 19 170000-area data block.
- Two data variables were added to the existing Asset Header block (MPM13): next event time variable and go\_flag variable.

- Two objects and their associated header files, nwrk.cpp and simnxt.cpp, were modified.
- The configuration database (CDB) was updated to manage HLA assets (The CDB holds instructions for linking external assets to JIMM.) Four instructions were added to the CDB file: TIME-FRAME-SIZE, LOOK-AHEAD, TIME-REGULATED, and TIME-CONSTRAINED. These CDB instructions can be used to define the asset's TM responsibilities and administrative conditions for each exercise.
- The HLA (and DSM) simulation architectures were modified to include the time-regulated and time-constrained operations available in the JIMM asset.
- The Language Data Base (LDB) was updated to extend JIMM to incorporate time management (The LDB holds core JIMM simulation language and code parsing instructions).

These changes give JIMM the ability to run time managed with CART and other entities in an HLA federation. In addition to the time management services, modifications were made to support the new CART-JIMM communication protocol that enables JIMM to process the data interactions described in the previous section.

### **3.2.3 *CART-JIMM Direct Shared Memory (DSM)***

#### **3.2.3.1 *CART-JIMM DSM Integration***

In addition to the HLA architecture, a Direct Shared Memory (DSM) architecture between CART and JIMM was implemented. In a DSM application, each computer within the system used to perform a multi-entity simulation contains its own shared memory card that is linked to the other computers in the system via a fiber optic ring. The shared memory cards used in the system for this project were the Systran Corporation SCRAMNet (RAMPlex Hardware Shared Memory; <http://www.systran.com/scmain.html>) cards. When data is written to a shared memory card on one computer, all the shared memory cards linked on the ring are updated (nearly) instantaneously. In theory, this type of network should allow the simulation to run extremely fast and eliminate delays due to message passing between entities and external network traffic.

The DSM architecture can be used in two ways in the CART-JIMM federation. First, if all simulation computers employ DSM, JIMM and CART could interact in the same fashion as in the HLA – that is, JIMM would obtain services from CART by subscribing to CART entities using data interactions. In this case, the HPM services would be available to any entity capable of subscribing over the DSM network.

Alternatively, HPMs could be dedicated to JIMM though shared memory while JIMM is simultaneously playing a part in a federation through HLA or another protocol, as shown in Figure 3-11. In this case, the other simulation tools would not have access to the HPMs. This configuration would allow the HPMs to provide services to JIMM quickly since communication would not be affected by network traffic or protocol delays in the federation.

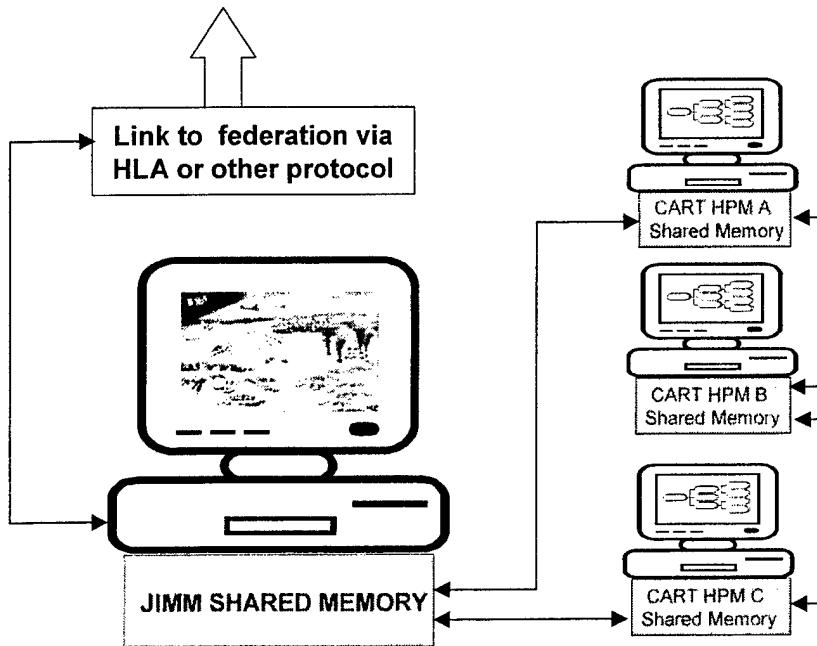


Figure 3-11. Alternate CART-JIMM direct shared memory configuration.

### 3.2.3.2 Direct Shared Memory Time Management Scheme/Executive

To implement the CART-JIMM client-server architecture in DSM, a time management scheme was necessary to manage the interactions between CART and JIMM entities. The DSM Constellation Executive was developed to provide the same functions for DSM configuration as the HLA RTI Time Management services provide for the HLA configuration. While the HLA supports a Federation, the counterpart in the DSM architecture is called a Constellation. The DSM Constellation Executive is linked to JIMM and CART entities through the shared memory card, as depicted in Figure 3-12.

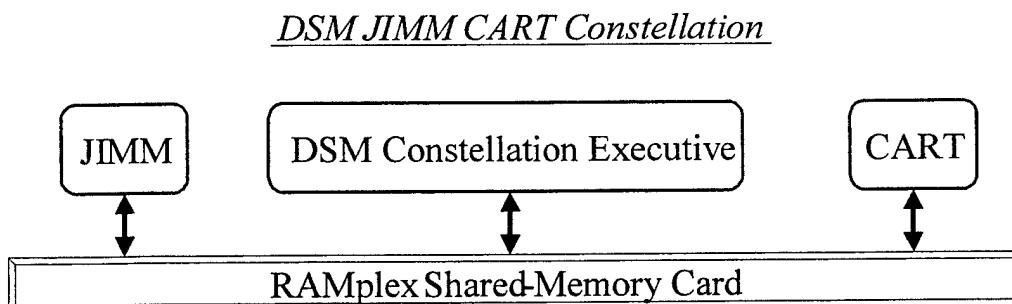


Figure 3-12. CART-JIMM Direct Shared Memory configuration.

The DSM Constellation Executive was created to perform three primary functions:

1. Allocating and configuring blocks of shared memory
2. Initializing client/server send and receive mailboxes in shared memory

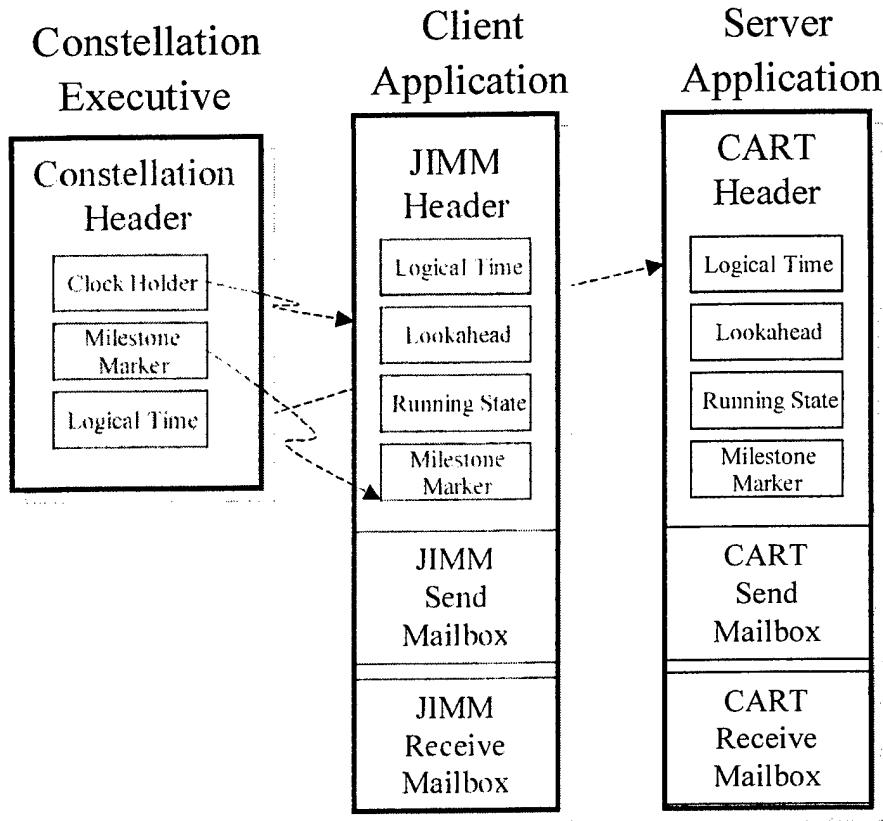
3. Providing Time Management (TM) Executive services. The DSM Constellation Executive has a TM Executive component that performs services similar to the HLA RTI TM services. The DSM TM Executive had to be created because there was no such software that could be leveraged

The following items were implemented in the design of the DSM Constellation Executive:

- Two primary layers/objects were added for migrating JIMM HLA to JIMM DSM: the Client Interface Direct Shared Memory and the SCRAMNet API.
- The shared memory was arranged so that constellation applications could determine which areas of shared memory are readable and writeable. They were arranged in ascending order for the hardware memory address as follows:
  1. Constellation Header
  2. Client Header
  3. Client Send Mailbox
  4. Client Receive Mailbox
  5. Server Header
  6. Server Send Mailbox
  7. Server Receive Mailbox

The layout of the TM Data Members with JIMM as the “client” and CART as the “server” are shown in Figure 3-13.

## DSM TM Data Members



**Figure 3-13. DSM Constellation Executive data members.**

The TM Executive and each application in the constellation contain TM related data variables. The TM Executive monitors these variables and adjudicates which application may proceed based on the logical time, the application that is the clock holder, and by propagating the milestone markers of each application.

### 3.2.3.3 Modifications to the CART Middleware for the DSM architecture

To enable the CART-JIMM client-server architecture, the capability to handle multiple entity subscriptions was added to the CART Middleware. For the DSM implementation, a DSM Link module was developed to support federation execution.

The communication protocol between CART and JIMM that was established for HLA also applies to the DSM implementation. The same definitions for data interactions are used in both HLA and DSM federations and are defined in further detail in Appendix B - Interface Control Document for Data Interactions between JIMM and the CART Middleware. For DSM, data interactions are passed between CART and JIMM entities by following a defined protocol for writing data to and reading data from the shared memory cards.

### 3.2.3.4 Modifications to JIMM to Enable the DSM Architecture

Modifications were made to JIMM to recognize and support the DSM Constellation Executive. Rather than receiving message via the HLA RTI, interactions are read from and written to the shared memory cards. As mentioned in the previous section, the format of the data interactions remains the same for DSM as for HLA.

## 3.3 CART-JIMM Client-Server Performance

### 3.3.1 Testing Overview

MA&D conducted performance testing to evaluate and compare the speed of each time management executive, especially the new DSM technology which, in theory, promised to offer a faster alternative to HLA. To test the performance of the HLA and DSM time management executives, MA&D designed a testing scheme measuring the duration, in real time, that each executive required to run 12 twelve minutes of simulation time for both a small scale and a large scale engagement scenario. The difference between the small and large scale scenario was the number of entities within the scenarios. The small scenario contained about 10 entities and the large scenario contained approximately 100 entities. The testing scheme configurations included three computers with the names, specifications, and applications shown in Table 3-1.

**Table 3-1. Test system configurations.**

Computer System Name	Processor Speed	Operating System	Memory	Applications used on system
CMCC1	800 MHz	Linux Redhat 7.2	384 MB RAM	JIMM
CMCC2	Pentium 4 1.4 GHz	Windows 2000	640 MB RAM	CART and Middleware
CMCC3	Pentium 4 1.4 GHz	Windows 2000	640 MB RAM	Any

MA&D set up three different testing scheme configurations. Each configuration placed the time management executive on a different computer to measure how performance related to computational power. The first configuration had the time management executive - HLA or DSM - located on CMCC1; the second had the time management executive on CMCC2; and the third had the time management executive on CMCC3. The first two configurations had the time management executive collocated with other applications (i.e. the time management executive with JIMM on cmcc1, and the time management executive with CART on CMCC2). The third configuration placed the time management executive on a dedicated processor without any other applications running.

MA&D calculated an average time and standard deviation after running and recording five trials for each configuration, time management executive, and scenario type (a total of 12 records). For each scenario, CART provided the human modeling for one Sector Operations Center (SOC), one Control and Reporting Center (CRC), and one Control and Reporting Post (CRP). The SOC, CRC, and CRP are command and control echelons found in a typical IADS. More information describing their behavior can be found in Appendix C, “CART Human Performance Model for IADS Scenario.”

### **Small scale scenario**

Scenario Description: Two hostile fighter aircraft approach a small IADS

IADS Details:

- One SOC
- One CRC
- One CRP
- Three SAM Brigades

### **Large scale scenario**

Scenario Description: 69 hostile aircraft approach a large IADS

IADS Details:

- One SOC
- Four CRCs
- Four CRPs
- Eight SAM Brigades

With the large scale scenario having more entities than the small scale scenario, the number of interactions between the federates increased significantly.

#### ***3.3.2 Small Scale Scenario Test Summary***

The system configuration and test results for the small scale scenario test are listed in Table 3-2 and Table 3-3, respectively. A summary of the test results are as follows:

Running in HLA:

- Averaged faster than real time in each of the three configurations.
- Performed optimally in configuration 1 with an average time of 8.5 minutes and real time factor of 0.71. The real time factor is an indicator of how fast the simulation ran compared to real time. For example, if a simulation required six minutes to simulate 12 real minutes of a mission, the real time factor would be 0.5 (i.e. the simulation executes in half the time it would take in real life). Having a real time factor of less than one indicates a simulation execution time faster than real time, whereas a real time factor over one indicates a simulation execution time slower than real time. As simulation speed increases, the real time factor decreases.

Running in DSM:

- Averaged slower than real time in each of the three configurations.
- Performed optimally in configuration 3 with an average time of 16.8 minutes and a real time factor of 1.4.
- Performance depended upon the computer location of the time management executive. When placed on a dedicated processor, the DSM time management executive was, on the average, more than three minutes (3 minutes, 18 seconds) faster than the next best configuration average – an improvement of 15.6%.

**Table 3-2. Small-scale scenario test system configuration.**

Applications	OS
CART Middleware	Windows only
JIMM	Linux only
HLA RTI	Linux or Windows
DSM RTI	Linux or Windows

Test machines			
	OS	Processor speed	RAM
CMCC1	Linux (RH 7.2)	800 MHz	384 MB
CMCC2	Windows 2000	Pentium 4 1.6 GHz	640 MB
CMCC3	Windows 2000	Pentium 4 1.6 GHz	640 MB

Test definition	
Wall Clock Start Time	Release from Time Manager at Simulation time = 6.0 sec
Wall Clock Stop Time	12.0 minutes Simulation Time
Execution Time	Wall Clock Stop Time - Wall Clock Start Time
Scenario	Small scenario, Iraq
Network	HLA - connected to LAN; DSM - disconnected from LAN

**Table 3-3. Small-scale scenario test results.**

	Test Machine			Performance (minutes)		
	CMCC1	CMCC2	CMCC3	Average	Std Dev	Real Time Factor
H	JIMM & HLA RTI	CART MW	not used	8.5	0.7	0.71
L	JIMM	CART MW & HLA RTI	not used	9.3	0.7	0.77
A	JIMM	CART MW	HLA RTI	9.8	0.8	0.82
D	JIMM & DSM RTI	CART MW	not used	20.8	1.9	1.73
S	JIMM	CART MW & DSM RTI	not used	19.9	0.4	1.66
M	JIMM	CART MW	DSM RTI	16.8	0.1	1.40

### 3.3.3 Large Scale Scenario Test Summary

The system configuration and test results for the large scale scenario test are listed in Table 3-4 and Table 3-5, respectively. A summary of the test results are as follows:

Running in HLA:

- Averaged slower than real time in each configuration.
- Performed optimally in configuration two with an average time of 91.8 minutes and real time factor of 7.65.
- Performance depended upon the location of the time management executive. When collocated with CART, the HLA time management executive averaged six minutes (6%) faster than the next best average time.

The increased amount of interactions in the large scenario drastically degraded performance relative to the small scenario.

Running in DSM:

- Averaged slower than real time in each configuration.

- Performed almost identically in each configuration, with its best in configuration 3 with an average time of 163.4 minutes and a real time factor of 13.61.
- Performance did not depend on the location of the time management executive. All three averages were within 1% of each other.
- The increased amount of interactions in the large scenario drastically degraded performance relative to the small scenario.

**Table 3-4. Large-scale scenario test system configuration.**

Applications	OS
CART Middleware	Windows only
JIMM	Linux only
HLA RTI	Linux or Windows
DSM RTI	Linux or Windows

Test machines			
	OS	Processor speed	RAM
cmcc1	Linux (RH 7.2)	800 MHz	384 MB
cmcc2	Windows 2000	Pentium 4 1.6 GHz	640 MB
cmcc3	Windows 2000	Pentium 4 1.6 GHz	640 MB

Test definition	
Wall Clock Start Time	Release from Time Manager at Simulation time = 6.0 sec
Wall Clock Stop Time	12.0 minutes Simulation Time
Execution Time	Wall Clock Stop Time - Wall Clock Start Time
Scenario	Large scenario, Iraq
Network	HLA - connected to LAN; DSM - disconnected from LAN

**Table 3-5. Large-scale scenario test results.**

Test Machine			Performance (minutes)		
cmcc1	cmcc2	cmcc3 - windows	Average	Std Dev	Real Time Factor
H JIMM & HLA RTI	CART MW	not used	99.3	0.6	0.71
L JIMM	CART MW & HLA RTI	not used	91.8	0.8	0.77
A JIMM	CART MW	HLA RTI	97.8	1.1	0.82
D JIMM & DSM RTI	CART MW	not used	164.4	0.9	1.73
S JIMM	CART MW & DSM RTI	not used	165.0	1.2	1.66
M JIMM	CART MW	DSM RTI	163.4	1.4	1.40

### 3.3.4 Performance Summary

Our tests show the following results:

- The small and large scenario simulations both ran faster in HLA than in DSM, indicating that HLA is likely a faster alternative to DSM for all configurations and scenario types. This result is highly contrary to what we expected since the key feature and reason for justifying the cost of DSM is its high speed capabilities. We believe a main contributor to this superior performance is the significantly greater level of effort that was used to implement Time Management services within HLA versus the limited level of effort that we could afford to implement Time Management services in DSM within the CMC2 program constraints.

- As the size of the simulation increases, the performance of both the HLA and DSM time management executives drastically decreased.
- DSM performs optimally for small scenarios on a dedicated processor.

Additional investigation and testing could determine if performance in HLA and/or DSM could be improved by employing alternative code implementations or other performance optimization methods.

## 4.0 IADS Model Development

### 4.1 Integrate Air Defense System (IADS)

A simulation that portrayed cultural effects on Command and Control (C2) within an IADS was developed to demonstrate the functionality of the new CMC2 tool and its use within the CART-JIMM client-server architecture.

A “generic” IADS C2 structure, shown in Figure 4-1, was used for the model. Each rectangle in the figure represents an entity that exists in JIMM, e.g. the SOC is a Sector Operations Center entity in JIMM. The entities that are shown as shaded boxes are entities that will subscribe to the HPM services developed with the CMC2 tool. A detailed description of the IADS entity models created with the CMC2 tool is provided in Appendix C, “CART Human Performance Model for IADS Scenarios.” Appendix C includes the description of each JIMM entity that is modeled (SOC, CRC and CRP) and the functions and algorithms that determine the behaviors of each model.

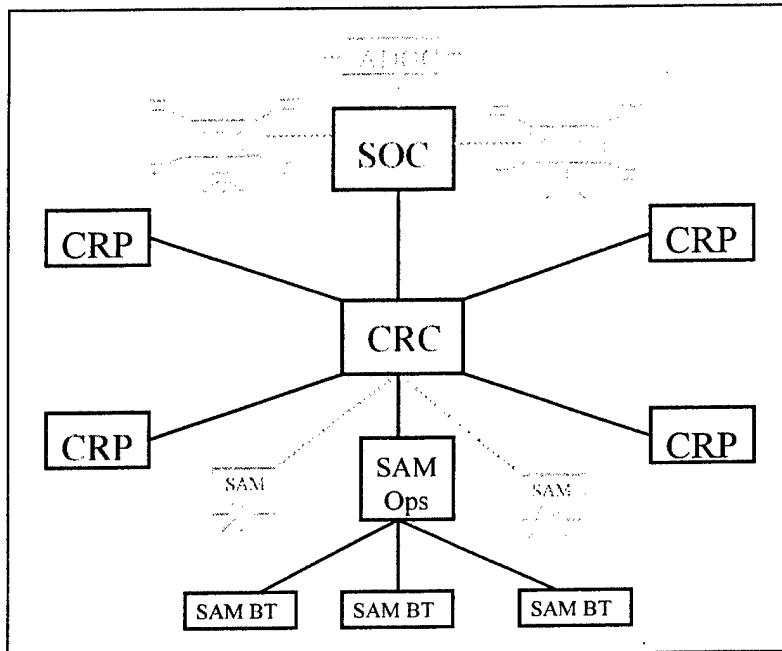


Figure 4-1. Generic IADS layout.

As shown in Figure 4-2, each JIMM entity is independently served by an HPM instance. The instances of each HPM entity only provides feedback based on the interactions with the subscribed JIMM entity – i.e. the CART HPMs do not affect each other and do not share information between each other.

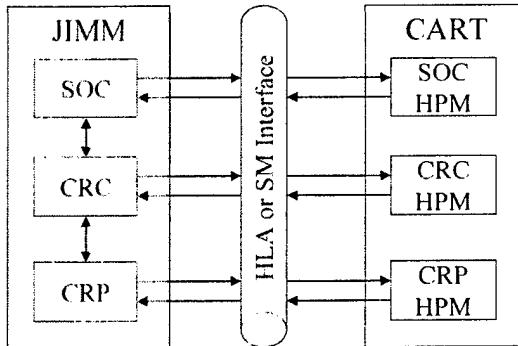


Figure 4-2. JIMM-CART interface.

## 4.2 IADS Scenarios

MA&D and Northrop Grumman designed three different mission scenarios for the integrated simulation: prewar, traditional, and SAMbush. Additionally, Northrop Grumman devised two different time periods in JIMM for the IADS hardware, one a current-day approximation and the other a forecasted future approximation. The main difference between the two approximations was the level of automation within the IADS hardware.

The following three sections give brief descriptions of each scenario. Appendix C, “CART Human Performance Model for IADS Scenarios,” provides more detail for each scenario and how the IADS reacts to each situation.

### *Prewar Scenario*

The prewar scenario finds the IADS in a precautionary posture as two enemy F-16s fly a nearby boarder patrol mission. The IADS in this scenario used a current-day time-period and included the following entities:

- One SOC
- One CRC
- One CRP
- Three SAM Brigades

Figure 4-3 shows a screenshot of JIMM for the prewar scenario. In the screenshot the CRP is shown sending track data to the CRC, indicated by the blue line connecting the two entities.

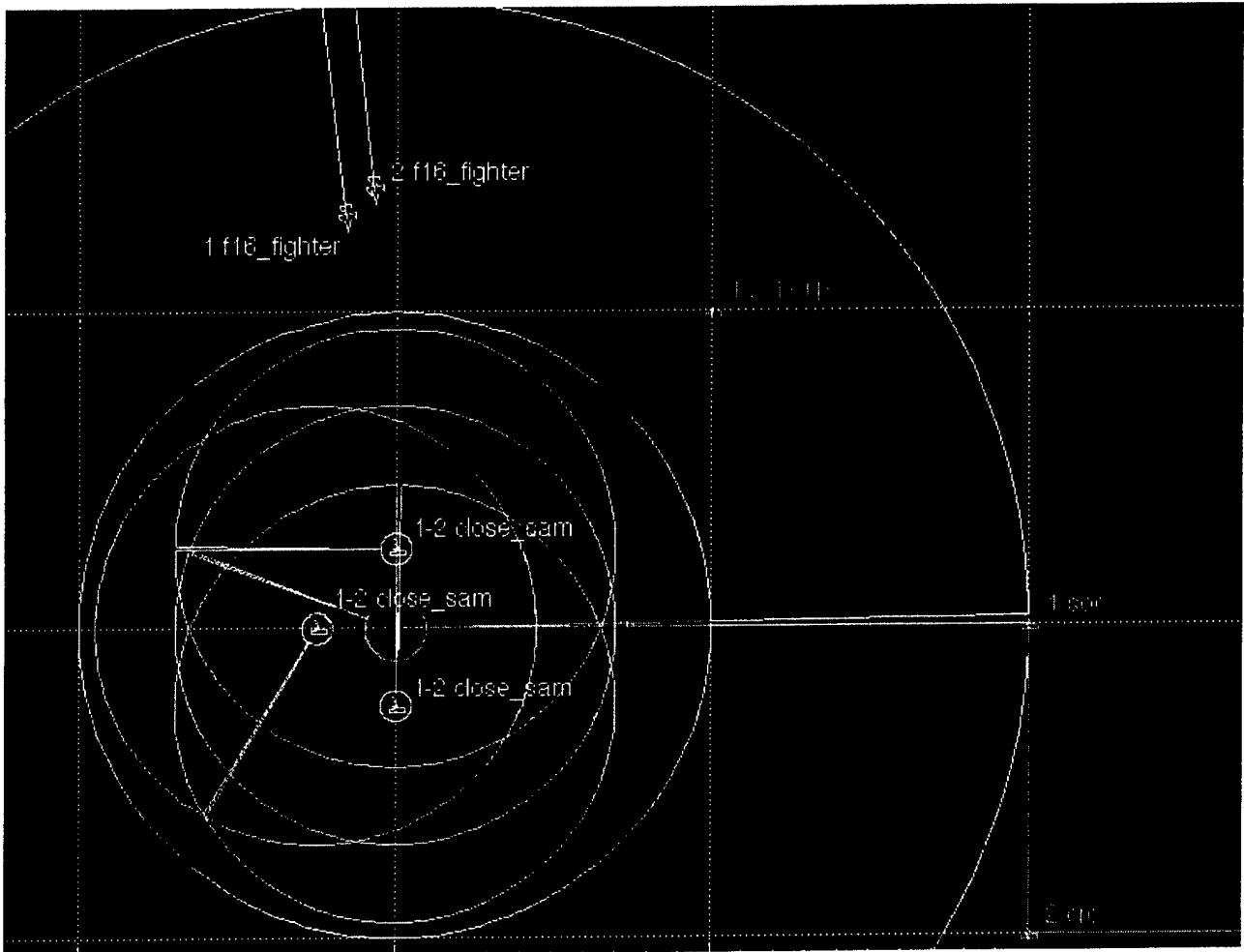


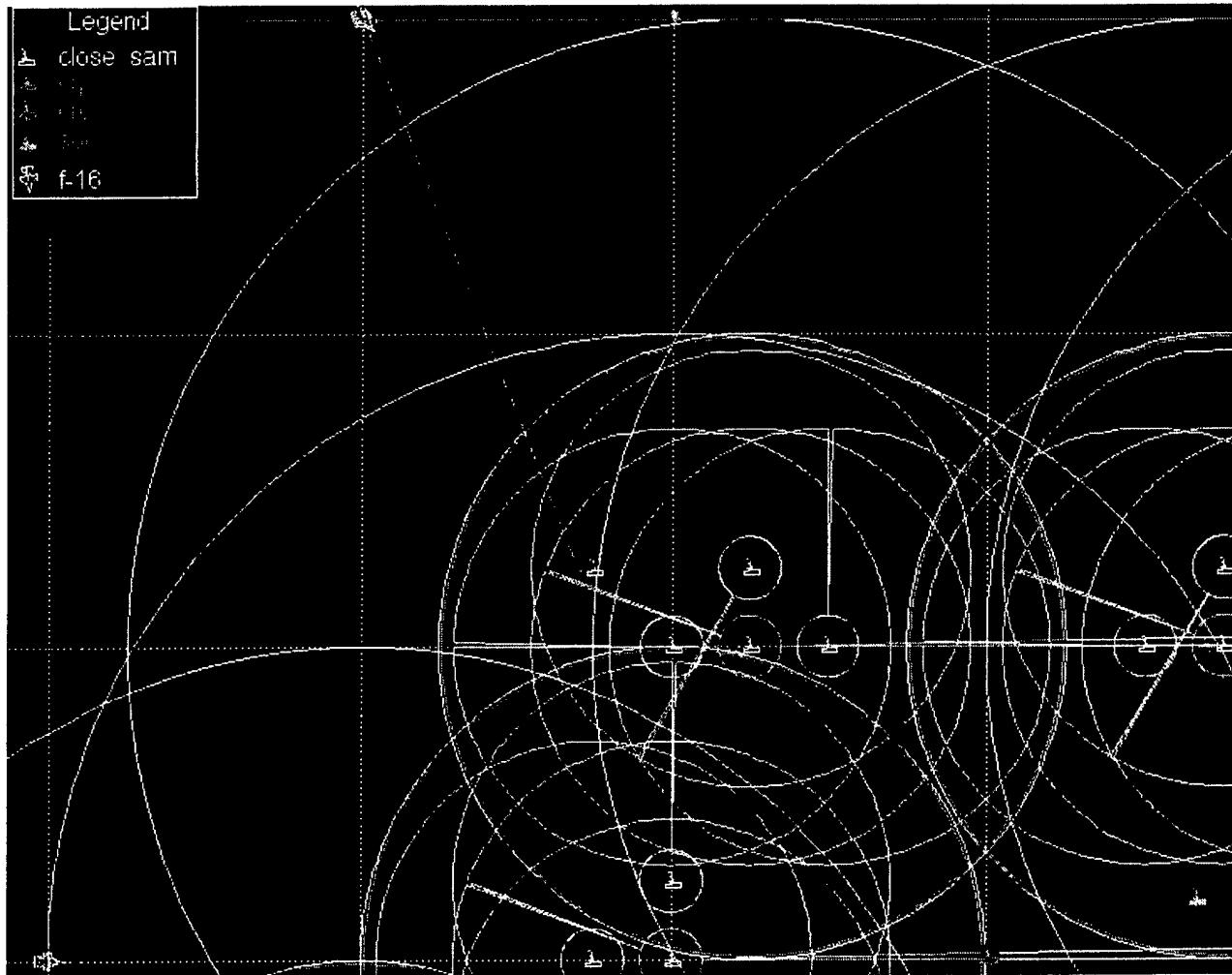
Figure 4-3. JIMM Prewar screenshot.

#### *Traditional Scenario*

The traditional scenario finds the IADS in a (full-blown) wartime situation defending against 69 invading aircraft. The IADS in this scenario used a future-day time-period and included the following entities:

- One SOC
- Four CRCs
- Four CRPs
- Eight SAM Brigades

Figure 4-4 shows a screenshot of a portion of the traditional scenario. Because of traditional scenario's size, this figure includes a legend in the top left hand corner to reduce text clutter. In the figure, a CRP has detected an F-16 with its radar, shown by the dotted purple line.



**Figure 4-4. JIMM Traditional Scenario screenshot.**

#### *SAMBush Scenario*

The SAMbush (SAM ambush) scenario finds the IADS in an unconventional wartime situation with the sole intention of using unexpected attacks to catch the enemy off guard. The layout of this scenario is similar to the one shown in Figure 4-3. The IADS in this scenario used a future-day time-period and included the following entities:

- One SOC
- One CRC
- One CRP
- Three SAM Brigades

#### **4.2.1 IADS scenario development requirements**

The implemented IADS system emulates a generic, yet realistic, threat weapon system. Each entity type within an IADS has been represented, and also its interrelationship with other entities. Each entity must be structured to interact with commanders and subordinate entities. The stimuli events that intrude on the IADS airspace are provided by JIMM. Decisions and responses of the IADS made as a result of those stimuli are adjudicated in the HPMs within CART.

#### **4.2.2 *Blue Force Representations***

Blue aircraft have been positioned and given a flight plan to stimulate the IADS systems. A simplification made in the construction of this simulation was that blue aircraft may shoot but cannot destroy IADS components. Because of this simplification, it was not necessary to implement detailed sensor, communication, and weapons systems for the blue aircraft. Similarly, there was no need for sophisticated jammer systems and blue communications networks for command and control. Also, onboard radar warning receivers were not used to trigger evasive maneuvers.

#### **4.2.3 *Red Force Representations***

Red force elements were created to represent communications networks and rule sets and tactics, as well as sensor and weapons. Some details include the following:

- Specific weapon systems were not implemented. However, platform definitions were parameterized. Parameters included how weapons are carried, which targets the weapons are effective against, Probability of Kill (PK) values, etc.
- The amount of targets a platform can track.
- Communication networks were created. Network characterization included landline, duplex, broadcast, etc.
- Engagement zones and areas of responsibility were attached to the IADS components.

The following assumptions were also applied to the red force representation:

- Equipment/resource assumptions:
  - Jamming was not considered.
  - No airborne interceptors (AIs) used.
  - Only radar-based sensors (EW, HF, FC) used. No visual, IR or other sensors.
  - No self-protection SAMS or AAAs used at any node.
  - Reliability/maintainability/sustainability not played.
- Scenario assumptions:
  - The SOC is invulnerable to damage or destruction.
  - No Red attrition of nodes served by human performance models.
  - SAM assets remain stationary during a scenario execution.
  - The SOC is not co-located with any of the assets under its control.
- Operations assumptions:
  - The SOC does not have direct control over any early warning radars.
  - The SOC does not have direct control over weapons or collocated dedicated weapons for self-preservation.
  - Intelligence at the SOC is received only through CRCs, i.e. no additional local intelligence sources.
  - Target assignment and cancellation must be acknowledged.

- No can't comply on cancel target

### 4.3 Cultural Factors for IADS Scenario

The function of the HPMs is to provide IADS entities with human performance behaviors that are based on cultural affects. Since no quantitative cultural data was available related to the behavior of these types of entities, a notional cultural model was created instead. As mentioned in Section 3.1.1, Cultural Modeling, three notional cultural factors were selected for the IADS model:

- 1) Distribution of Power
- 2) Willingness to Take Risk
- 3) Familiarity with the Enemy

Although cultures could be rated on a continuum for each of these factors, this model limited the values to "high" or "low" for simplicity. *These factors and their effects are notional for the model created.* However, data may be found in the future to substantiate them. For these IADS scenarios, they are used with the following definitions.

**Distribution of Power (DP)** is the perceived difference in power between an individual (member of the military) and his superior. A small (or "low") DP indicates that the individual operates in a traditionally strict environment where subordinates follow orders as passed down from their superiors and rarely question authority. Large (or "high") DP indicates that the individual operates in an environment that is more egalitarian – one in which he is empowered to make decisions if and when necessary, and may even question authority. As an example of the effects of DP, an officer in a culture with low DP may hesitate to make decisions, or will make no decisions, when contact is lost with his superior. On the other hand, in a culture with high DP, the officer readily accepts responsibility and control of the situation until communication is reestablished.

**Willingness to Take Risk (WR)** is defined for this project as an individual's willingness to make decisions that place him in vulnerable situations thereby risking the consequences of errors. An individual with high WR will make decisions that may likely place him in a vulnerable position. He is willing to proceed with attacks on an enemy even when the reliability of intelligence data and/or predicted outcome is unknown. He is also willing to place himself at higher risk in a situation in order to achieve potentially higher gains. An individual with low WR will hesitate to proceed with actions against an enemy until he is satisfied that his actions will produce a desirable outcome. He is conservative in his actions and is concerned with making errors. As a contrast, a commander in a culture with high WR would be more likely to fire upon an unknown aircraft, risking a civilian casualty, than one with low WR. He may also lure in opposing forces by allowing them closer access to a high value target thereby increasing his chances of a successful ambush.

**Familiarity with the Enemy (FE)** is defined as the extent to which a culture has had prior interaction with its enemy. A high FE indicates that the culture has had significant interaction with the enemy culture and is familiar with many aspects of that culture, particularly (in this case) with their warfare methods and strategies. A low FE indicates that the culture has had minimal interaction with the enemy culture and is unfamiliar with their warfare methods. As an example, a culture with high FE will have more experience and, likely, better success when engaging their enemy. A culture with low FE may have difficulty determining the best strategy to employ and, therefore, be less successful, or less efficient, when engaging their enemy.

#### 4.4 IADS Cultural Demonstration Results

IADS demonstration results showed how the two countries of interest, Iraq and North Korea, behaved differently under identical conditions in three different scenarios. General differences included the order and selection of assigned aircraft, the respective times of assignment, and the locations of the aircraft when assigned. Exact differences, detailed in the data tables below, were dependent on the scenario executed. The next few paragraphs provide a brief description of the threat evaluation algorithm that the IADS employs, the WR cultural factor's influence on the algorithm, and the chosen WR cultural factor values. The sections that follow will provide more detailed information for each scenario. Appendix C, "CART Human Performance Model for IADS Scenario," provides greater detail of all the HPM algorithms.

The threat evaluation algorithm gives threat scores to aircraft detected by the IADS radar. Calculating this score provides the IADS with an indication of the level of threat of an aircraft. Five evaluation parameters help quantify the inherent threat of aircraft: aircraft type, speed, (X, Y) location, altitude, and directional heading. The calculated value for the threat score of an aircraft is then compared to the "track assignment cutoff level," – a nominal threshold value representing the demarcation of when an aircraft becomes detrimental to an IADS.

Military history and consultation with several SMEs regarding the behaviors of North Koreans and Iraqis lead to the assignment of a high willingness to take risk (WR) for Korea and a low WR for Iraq. The United States past history of military engagements with North Korea reveals that it is a very structured and disciplined country whose combatants follow orders closely and intrepidly serve their country. Iraqis, conversely, have demonstrated opposite behaviors. At the beginning of the first Gulf War, Iraqi SAM operators behaved as expected, but as the war continued they soon learned that turning on their radar or firing a missile was tantamount to suicide as their positions were quickly compromised to U.S. intelligence. As a result, the Iraqi IADS system became ineffective because SAM operators would inefficiently fire at long ranges with low probability of kill (PK) levels to avoid becoming a U.S. target.

Relating the cultural factor to threat evaluation, North Korea (with a high WR) gives lower scores than Iraq (with a low WR) gives to aircraft at identical distances (based on (X, Y) location) from the IADS<sup>1</sup>. The WR factor is the only cultural factor that differs between the two countries for these scenarios, i.e. MA&D gave North Korea and Iraq identical values for DP and FE. For these models, (X, Y) location is the only portion of the threat evaluation process influenced by the cultural factors. Each culture scores the remaining four evaluation parameters identically. (Note: In these scenarios, MA&D chose to have culture influence only one evaluation parameter to keep the models simple. However, the CMC2 tool does not limit the number of parameters used in the model. One can adjust the threat evaluation algorithm or any other area where culture affects the model as seen fit for the purpose of the study). The lower evaluation scores in the North Korea model results in a threat score that is lower than the threat score that would be generated by the Iraq model in response to the same situation. Since the threat score determines when the IADS will make a track assignment, the resulting behavior of the North Korea model is to allow aircraft in closer proximity to the IADS before making a track assignment while the Iraq model will make track assignments earlier when the aircraft are farther away.

---

<sup>1</sup> The values selected for the threat evaluation process are purely notional and intended to serve as a working example of how a country or culture might behave. These notional values reflect general descriptions of the two cultures from SME and literature sources for the sole purpose of demonstrating the functionality of the CMC2 tool and are not based on actual data.

### *Traditional Scenario Differences*

In traditional scenarios, the IADS gives an assignment to all invading aircraft that exceed the nominal track assignment cutoff level (a value of 750). Table 4-1 and Table 4-2 show assignment samples for each culture under an identical scenario. The scenario consists of two F-16 fighters that are approaching the IADS. (This scenario is a simplified version of the traditional scenario described in Section 4.2, IADS Scenarios.) A careful analysis reveals that culture can affect IADS assignment behavior in the following areas: order of assignment, time of assignment, and location of aircraft at time of assignment. (Note: Fire authorization, in the last column in the data tables, is the type of assignment sent from the SOC. Its value becomes important in the prewar scenario described briefly in the next section and in depth in Appendix C). Because Iraq gives higher scores to aircraft than North Korea at identical locations, aircraft for an Iraqi IADS typically exceed the cutoff level earlier in the scenario causing Iraq to assign aircraft at earlier times and at longer distances from the IADS. Take for example Track ID 11. Iraq assigns Track 11 first at time 3.20 and at a distance of 165.21 km from the IADS. North Korea assigns the same track second at time 4.88 and at a distance of 124.97 km from the IADS.

**Table 4-1. Iraq assignment samples from a traditional scenario.**

Assignment Time (min)	Track ID	Threat Score	Distance From IADS (m)	X Position (m)	Y Position (m)	Speed (m/s)	Heading	Altitude (m)	Fire Authorization
3.20	11	830	165210	-220076	-136014	439	173	1500	1
4.08	12	950	151359	-208111	-148856	429	173	2500	1

**Table 4-2. North Korea assignment samples from a traditional scenario.**

Assignment Time (min)	Track ID	Threat Score	Distance From IADS (m)	X Position (m)	Y Position (m)	Speed (m/s)	Heading	Altitude (m)	Fire Authorization
4.08	12	770	146741	-207658	-153459	422	173	2500	1
4.88	11	770	124971	-215522	-175996	401	173	1500	1

### *Prewar Scenario Differences*

The prewar scenario finds the IADS in a precautionary posture as two enemy aircraft fly a nearby boarder patrol mission. The purpose of the prewar scenario was to demonstrate how some cultures will “light up” a non-threatening aircraft (with a threat score below the track assignment cutoff level) to provoke it into retreating or possibly into attacking the IADS. To “light up” an aircraft means to acquire an aircraft with SAM targeting radar feinting an imminent SAM but with no real intention of weapon release. MA&D chose to have North Korea, with a high WR, be the culture to “light up” non-threatening aircraft, an unnecessary risk for an IADS. Appendix C provides further detail on the prewar track assignment algorithm.

Table 4-3 and Table 4-4 show assignment samples from each culture in an identical prewar scenario. An analysis of the tables reveals that North Korea “lit up” (fire authorization of zero) the two aircraft very early in the simulation and later assigned the two aircraft after becoming threatening with full weapons release authority (fire authorization of one). Iraq never “lights up” the two aircraft and instead waits until they become threatening to assign the aircraft with full weapons release authority.

**Table 4-3. Iraq assignment samples from a prewar scenario.**

Assignment Time (min)	Track ID	Threat Score	Distance From IADS (m)	X Position (m)	Y Position (m)	Speed (m/s)	Heading	Altitude (m)	Fire Authorization
2.3	11	950	150582	-220299	-150792	448	169	1500	1
2.4	12	950	151586	-241405	-154178	449	169	2500	1

**Table 4-4. North Korea assignment samples from a prewar scenario.**

Assignment Time (min)	Track ID	Threat Score	Distance From IADS (m)	X Position (m)	Y Position (m)	Speed (m/s)	Heading	Altitude (m)	Fire Authorization
0.38	12	530	196349	-248348	-109696	505	169	2500	0
0.57	11	530	199146	-227826	-102807	500	169	1500	0
2.05	12	770	150215	-241212	-155549	449	169	2500	1
2.23	11	770	153298	-220748	-148112	448	169	1500	1

#### *SAMBUSH Scenario Differences*

The SAMbush (SAM ambush) scenario finds the IADS in an unconventional wartime situation with the sole intention of using unexpected attacks to catch the enemy off guard. Enemy aircraft flying in the air sector believe that the IADS has been compromised and get a false sense of security. After the IADS radar has detected a pre-determined number of aircraft, the SOC orders a track assignment with fire authorization. North Korea, having a high WR, will wait until the IADS has detected four aircraft before issuing an assignment; Iraq will wait to detect two aircraft. By allowing more aircraft to fly into the IADS sector, North Korea runs the risk of suffering more damage than Iraq. Below, Table 4-5 and Table 4-6 show assignment samples from each culture in an identical SAMbush scenario. An analysis shows that Iraq makes a track assignment earlier in the scenario than North Korea and to a different aircraft.

**Table 4-5. Iraq assignment samples from a SAMbush scenario.**

Assignment Time (min)	Track ID	Threat Score	Distance From IADS (m)	X Position (m)	Y Position (m)	Speed (m/s)	Heading	Altitude (m)	Fire Authorization
0.47	102	945	100000	-300000	-300000	200	90	3000	1

**Table 4-6. North Korea assignment samples from a SAMbush scenario.**

Assignment Time (min)	Track ID	Threat Score	Distance From IADS (m)	X Position (m)	Y Position (m)	Speed (m/s)	Heading	Altitude (m)	Fire Authorization
0.53	98	465	223606	-300000	-500000	200	90	3000	1

## 5.0 Accomplishments

The CMC2 program accomplished numerous objectives. First, a toolset was developed that allows an analyst to build Human Performance Models (HPM) and include cultural factors that can affect decisions and performance parameters generated by the HPM. This toolset includes the capability for the analyst to compose cultural factors and apply these factors through the inclusion of cultural macros within the HPM. These capabilities allow the analyst to use the HPM to predict decisions and performance parameters.

The second objective was to develop the capability for HPMs to interact with the Joint Integrated Mission Model (JIMM) constructive simulation through both a High Level Architecture (HLA) interface and a direct shared memory (DSM) interface. This interconnection capability allows high fidelity HPMs to impact entity behaviors within JIMM and provide better behavioral representation for the entities.

The last objective was to utilize the CMC2 toolset with JIMM in the Integrated Air Defense System (IADS) domain. HPMs of the nodes within an IADS system for two cultures were developed using the CMC2 toolset. The entities and scenarios for the IADS were developed in JIMM. Using the client-server architecture, the HPMs were able to influence the performance of the IADS nodes during the simulation.

### **5.1 Follow-on work**

To leverage the accomplishments of this project, we made recommendations to pursue the following activities.

1. Transition the JIMM HLA interface to the JIMM user community at the earliest convenience. The time management enhancements to JIMM developed during this project would be very beneficial to the entire JIMM community. Additionally, since modifications were made in JIMM, additional testing and documentation need to be performed to allow these modifications to become part of the JIMM baseline version.

MA&D and Northrop Grumman Mission Systems (NGMS) have submitted a proposal to complete the integration of the JIMM modifications for CMC2 into the JIMM baseline version – the version that is formally released to the JIMM community.

2. A powerful capability has been created under the auspices of this effort. Namely the combination of a detailed human performance model with an industry accepted and widely used mission level model. The power of this capability extends well beyond cultural factors modeling to include the capacity to model human factors effects on mission performance and should be marketed to other potential users. This capability has potential to benefit the acquisition community for design tradeoff studies focusing on topics such as “potential improvements resulting in increased mission effectiveness.” Additionally, the training community’s simulation tools could be enhanced to account for human behaviors and their effects on opposing forces or on the trainees themselves.

MA&D has pursued several potential users including the National Air Intelligence Center (NAIC), Electronic Systems Center (ESC), and Air Intelligence Agency (AIA). Initially, interest was expressed by NAIC to integrate the CMC2 tool with Ternion’s FLAMES application, one of the applications used by analysts at NAIC. However, after several meetings between MA&D, NAIC, and Ternion to discuss the design requirements and the details of the integration, NAIC decided not to pursue this integration. Consequently, AFRL cancelled this effort.

### **6.0 Acknowledgements**

The authors would like to thank Geert Hofstede for his invaluable contributions to the field of cultural anthropology and providing the inspiration for the CMC2 cultural parameters. Helen Klein, Anna Pongonis, and Gary Klein also deserve recognition and thanks for their work on cultural barriers in a multinational C2 environment.

Furthermore, the authors would like to express appreciation to NAIC for providing an unclassified functional description of an IADS. This document, in particular, proved to be invaluable during the creation of the IADS models within the CART HPM.

## 7.0 References

- [1] Nils D. LaVine and Steven D. Peters: "*An Advanced Software Architecture for Behavioral Representation within Computer Generated Forces*" Final Report, Defense Modeling and Simulation Office, Alexandria, Virginia, December 2001.
- [2] Helen Altman Klein, Anna Pongonis and Gary Klein: "*Cultural Barriers to Multinational C2 Decision Making*" Command and Control Research and Technology Symposium, Naval Postgraduate School, Monterey, CA, June 2000.
- [3] Hofstede, Geert H.: "*Culture's Consequences: International Differences in Work-Related Values*", Sage Publications, Beverly Hills, 1984.

## **Appendix A: Joint Integrated Mission Model (JIMM) Design Document for Implementing Time Management**

The following report documents the implementation and testing of the JIMM Time Management services.

# **Joint Integrated Mission Model (JIMM)**

## **Design Document**

### **For Implementing Time Management**



Contract Number:  
GS-35F-4522-G

18 November 2002

Revision 1 – 14 April 2003

*Prepared and Submitted to the*  
**U.S. Air Force Electronic Systems Center**



ESC/CXCM  
45 Arnold St.  
Hanscom AFB, MA 01731-2142

*Prepared by*  
**NORTHROP GRUMMAN**  
*Mission Systems*

6001 Indian School Rd. NE  
Albuquerque, New Mexico 87110

## Foreword

This *JIMM Design Document for Implementing Time Management* has been prepared as a deliverable under Contract GS-35F-4522G by the technical staff of Northrop Grumman Mission Systems, Inc., Albuquerque, New Mexico. This document is submitted in compliance with the Statement of Work for Delivery Order 248.

Reviewed and approved by:

---

Craig Jones  
Northrop Grumman Mission Systems Project Lead  
Northrop Grumman Mission Systems

# Table of Contents

<b>FOREWORD .....</b>	<b>39</b>
<b>TABLE OF CONTENTS .....</b>	<b>40</b>
<b>1. REQUIREMENT OVERVIEW.....</b>	<b>41</b>
<b>2. INTRODUCTION.....</b>	<b>43</b>
2.1    EXECUTION USING RTI TIME MANAGEMENT IN HLA MODE .....	43
2.2    EXECUTE USING TIME MANAGEMENT EXECUTIVE IN DSM MODE.....	46
<b>3. PLANNED JIMM IMPLEMENTATION .....</b>	<b>46</b>
3.1    DATA STRUCTURES .....	49
3.2    JIMM INPUT.....	50
3.2.1 <i>JIMM Input Syntax</i> .....	50
3.2.2 <i>Target Routines</i> .....	52
3.3    CODE DESCRIPTION.....	52
3.3.1 <i>Target Routines</i> .....	52
3.3.2 <i>Algorithmic Changes</i> .....	52
3.4    OUTPUT DESCRIPTION.....	52
<b>4. TESTING APPROACH .....</b>	<b>52</b>
4.1    HLA TIME MANAGEMENT SERVICES .....	52
4.1.1 <i>HLA Time Management, Baseline Case</i> .....	53
4.1.2 <i>HLA Time Management, Look-Ahead Input Case</i> .....	53
4.1.3 <i>HLA Time Management, Different Look-Ahead Case</i> .....	53
4.1.4 <i>HLA Time Management, Time Regulated Case</i> .....	54
4.1.5 <i>HLA Time Management, Time Constrained Case</i> .....	54
4.1.6 <i>HLA Time Management, Frame-Based Case</i> .....	54
4.2    DSM TIME MANAGEMENT SERVICES.....	54
4.2.1 <i>DSM Time Management, Baseline Case</i> .....	54
4.2.2 – 4.2.6 <i>DSM Time Management, Other Cases</i> .....	55

# 1. Requirement Overview

The Joint Integrated Mission Model (JIMM) is a general purpose, data-driven, conflict simulation/environment generator. Each of the words in the previous sentence contains an enormous amount of information about JIMM and its capabilities.

**General purpose** – refers to the capability in JIMM to describe conflicts across a broad range of issues, detail, geographical extent, and time. General purpose also means that JIMM has a balanced approach to all aspects of the conflict. JIMM internally does not know anything about naval (or space, or army, or political, or environmental) issues, nor does it need to know anything about these issues. JIMM does not know anything about military forces, political organizations, or whatever issues are important to the decision maker. It is up to the user to specify these issues to the amount of detail that he/she thinks is sufficient to address them.

**Data-driven** – refers to the fact that each entity in the scenario is specified through parameters, interactions, and interrelationships defined by the databases. JIMM internally only knows about a few generic types of processes: moving, thinking, sensing, communicating, shooting, and disrupting or jamming. JIMM is a generic modeling environment without hard-coded entities of any kind. Every entity is a combination of systems, tactics and susceptibilities represented by these generic processes.

**Conflict** - refers to situations in which there is some contention over the use or control of resources. The contention could be within someone's mind, or it could be a widespread conflict among several factions. The nature of the conflict can change over time. The conflict might start as a relatively peaceful discussion between the participants and escalate to a highly energized, lethal dispute. JIMM contains very few internal assumptions about the nature of the conflict.

**A simulation** - represents the changes in the conflict over time. A simulation is dynamic, not static. Thus, the user needs to set up the initial conditions for the simulation and the basic assumptions or rules in JIMM about cause and effect that drive the changes in the conflict as time progresses.

**Environment Generator** - refers to the fact that JIMM performs an important function that is missing from many other simulations. In fact, the environment generation aspect contributes to the “completeness” of JIMM. Environment generation refers to the need (and ability) to provide realistic backdrops for other computer simulations, human-in-the-loop simulators, or hardware-in-the-loop stimulators. This feature allows the user to take advantage of higher fidelity resources that otherwise would not be utilized in an integrated representation of a conflict. JIMM has been used to represent a full spectrum of simulated entities or players in distributed networks with other simulations, simulators, hardware, and human-in-the-loop operators. JIMM has also been used to represent aspects of the simulated environment not represented by other components in a network or model federation. For example, manned flight simulators, operating in a virtual environment provided by JIMM, have been used to represent developmental and/or existing aircraft. Missile simulators or highly detailed missile simulations have been integrated with JIMM resulting in a higher fidelity representation of some Surface-to-Air missile units. Human-in-the-loop decision makers have operated in a JIMM virtual environment at command posts or control centers receiving updates to the battlefield situation and directing constructive, virtual, or live forces. While these examples have executed in real time, constructive analyses have been conducted using JIMM integrated with other computer simulations at faster than real time.

JIMM has DIS and HLA interfaces as well as the shared memory interface used to link to a variety of simulators, stimulators and simulations mentioned above. JIMM will interface with the Combat Automation Requirements Testbed (CART) by using both the HLA interface as well as a

direct, shared memory approach. A high level depiction of the JIMM architecture and capability is shown in Figure 1.1.

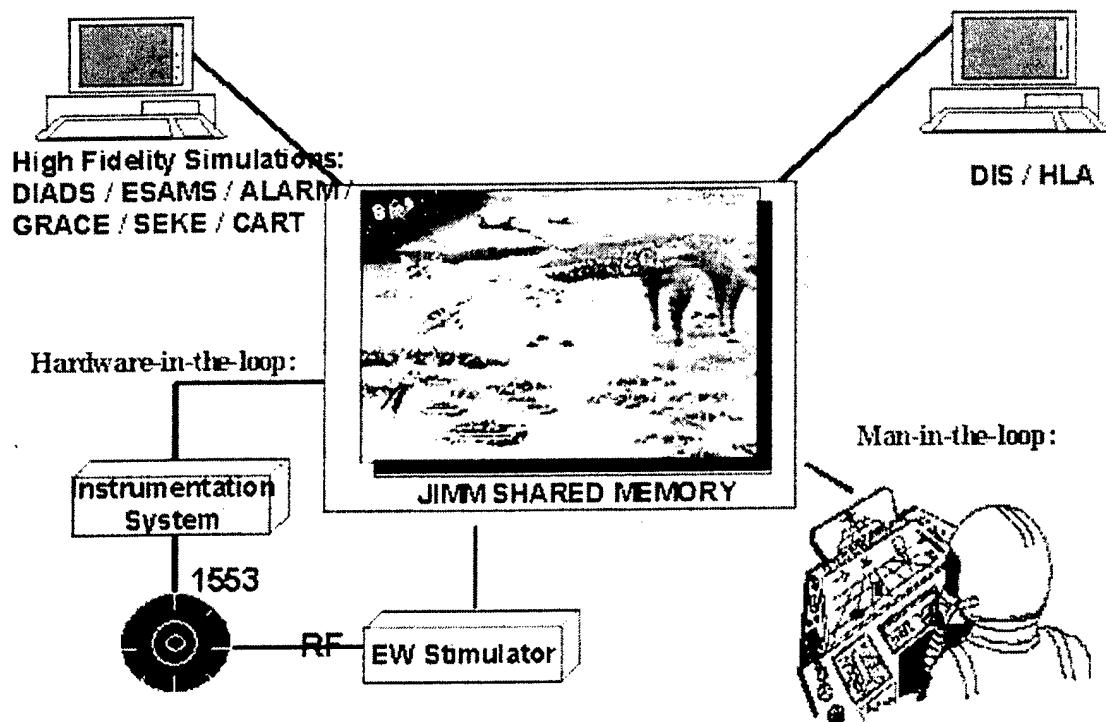


Figure 1.1. JIMM Architecture Overview.

The shared memory interface is well-defined and has been used by military analyst and software developers for over 15 years to link JIMM and its predecessors to hardware, simulators, computer simulations, and distributed networks. The shared memory interface contains position and kinematics data for all platforms that are in the virtual environment, including those controlled by external assets (referred to as "ghosting"), and it also contains other scenario related data. Assets, including JIMM, communicate with each other by means of mailbox messages located in shared memory. These messages contain data related to events such as weapon firing, assignments, results of sensing opportunities, and system status changes as well as a wide range of other information. For example, the Fighter Requirements Evaluation Demonstrator (FRED) cockpit simulator has been linked to JIMM by means of an HLA interface as well as a direct, shared memory interface. FRED controls the position of the associated "ghost" aircraft in the virtual environment, engages targets and fires weapons, and manages sensor and other systems. JIMM supplies FRED with sensing information from the sensors on the "ghosted" aircraft. Additionally, assets can generate and receive communications over the virtual networks in the scenario through the shared memory interface.

In addition to the simpler functions of movement, sensing, terrain occulting, and weapons firing, JIMM incorporates concepts such as perception, command and control, communication, jamming, reactive logic, planning, and the creation and absorption of players and resources. These concepts will be expanded upon and finely tuned to meet requirements of specific CART modeled entities in a combined simulation through the CART/JIMM Interface. JIMM is also a highly flexible, real-time threat generator and run time executive that supports exercises containing constructive, virtual, and live players. JIMM permits the substitution of any constructive system, platform, or player by its virtual or live counterpart. CART models will

replace human thinkers in the JIMM virtual environment to provide high fidelity, human performance modeling.

JIMM is an event-driven simulation in which all events are ordered through a single event queue. This queue is the main runtime bottleneck. A challenge that will be faced when incorporating the “middleware” scheme of CART into the architecture of JIMM will be to maintain maximum federation execution speed while also keeping models in sync with each other. Both HLA Time Management services and a direct, shared memory time management scheme will be employed to accomplish these goals.

This capability relates to Section 3.1 of the Joint Accreditation Support Activity (JASA) Accreditation Support Packages (ASP), Volume II for JIMM.

## 2. Introduction

The implementation of time management into JIMM breaks into two logical pieces: use of HLA Time Management Services and a direct, shared memory (DSM) interface. The majority of the differences in the implementation will be in the two interfaces. The JIMM side will be very similar. The time management executive is a separate process that performs an administrative function allowing JIMM and other assets to maintain time synchronization. The JIMM-to-CART interface (J2C) is being developed as a JIMM HLA asset and is a separate executable. The J2C shall execute in a time-managed manner to facilitate time synchronization between JIMM and CART object interactions. The J2C shall allow a cause and effect relationship to exist where the temporal aspects of federation execution have an accurate temporal effect between the CART federate and JIMM federate/asset. Part of the design will be common to both the HLA and DSM modes.

An asset will store the time of the next event for that asset in a common location in shared memory. For JIMM, that will be just the time of the next event. For other assets that might be the next event time, the minimum of all of the next event times, the next frame time, maximum game time (or MAX\_FLOAT) if no events are scheduled, etc. Except for the HLA interface asset, all assets will update their next event time. All assets will then wait for a time advance grant. Once an advance grant is given and processed, all assets will re-evaluate the time of their next event and post. The process then repeats.

### 2.1 Execution using RTI time management in HLA mode

In the HLA mode, the JIMM HLA interface will use the RTI time management services to assure temporal relationships are maintained between activities modeled in JIMM and the human performance models (HPM) in CART. This capability shall allow for the analysis and repeatability of a J2C federation execution. *Note: The CART emulator, enhanced to use HLA time management services (i.e., Time Constrained and Time Regulated), will be used to test the enhanced JIMM HLA interface.* There are two parts to the JIMM HLA interface upgrade to handle time management. First, upgrade the HLA interface to incorporate the HLA Time Management Services; second, create a new module that manages JIMM and all the other assets as a single federate. This new module will perform the following functions using the HLA Time Management Services:

The JIMM federate produces time-stamped events, which its Local RTI Component (LRC) communicates to the CART federate.

**enableTimeConstrained()****ABSTRACT**

This service instructs the LRC to constrain the advancement of the federate's time based on the federation's time, and to deliver time-stamp ordered events in the correct order.

**enableTimeRegulation()****ABSTRACT**

This service instructs the federation to consider the federate's logical time for the purposes of governing the advancement of federation logical time.

**nextEventRequest()****ABSTRACT**

This service advances the federate's logical time to the time-stamp of the next relevant time-stamp-ordered event in the federation. A **timeAdvanceGrant()** federate-ambassador callback will occur after one or more time-stamp-ordered events have been delivered or the federation lower-bound time stamp (LBTS) advances past the specified cutoff time.

**queryFederateTime()****ABSTRACT**

This service is used by the JIMM federate to obtain its current logical time.

**queryLBTS()****ABSTRACT**

This service is used by the JIMM federate to obtain the current federation lower-bound time-stamp.

**queryLookahead()****ABSTRACT**

This service is used by the JIMM federate to obtain the size of the internal extending forward from the federate's logical time at a given point in execution in which the JIMM federate will not generate any time-stamp-ordered events.

In order to synchronize the federates, a temporary "dummy" federate will be used to create the federation and holds logical time to zero until all federates are ready. Once all federates are ready, the dummy federate resigns and destroys itself.

The HLA Time Management Services will adjudicate the following:

- Establish or associate events with JIMM federate time

- Regulate interactions, attribute updates, object reflections or object deletion by federate time scheme
- Support causal behavior within a federation

In general, time advances must be coordinated with object management services so that information is delivered to each federate in a causally correct and ordered fashion.

The JIMM federate is "time regulating" and may associate some of its activities (e.g., updating instance attribute values and sending interactions) with points on the federation time axis. Such events are said to have a "time-stamp." Both the JIMM and CART federates are interested in discovering events in a federation-wide, time-stamp order and are said to be "time constrained."

The JIMM federate will produce time-stamped events, which its Local RTI Component (LRC) communicates to the CART federate. The LRC of CART orders all arriving time-stamped events by the time at which the events are said to occur. The time management functions of these LRCs are listed in Section 2.

The JIMM federate will generate time-stamp-ordered (TSO) events. TSO events are said to occur at a specific point in time. CART will also be time regulated. Therefore the CART federate will also generate TSO events. The JIMM federate perceives the current time to be "*tcurrent*." JIMM will not dynamically alter its status from regulating to non-regulating dynamically (i.e., "on-the-fly"). The JIMM federate promises that any TSO events it generates will occur equal to and no earlier than "*tcurrent* + *tlookahead*." The look-ahead value *tlookahead* represents a contract between the regulating federate and the federation. It establishes the earliest possible TSO event the federate can generate relative to the current time, *tcurrent*. The look-ahead default value will be 0.0001 seconds at the time it becomes regulating. The JIMM federate will not alter the look-ahead value dynamically. To avoid special conditions and restrictions JIMM will not use look-ahead value equal to zero. All TSO events will occur at a time "*tcurrent* + *tlookahead*" or greater. The CART federate has an associated Lower Bound Time Stamp (LBTS). The LBTS specifies the time of the earliest possible time-stamp-ordered event the federate can receive. The LBTS is determined by looking at the earliest possible message that might be generated by the JIMM federate. LBTS changes as JIMM advances in time. The CART federate cannot advance beyond its LBTS.

JIMM will federate with either Event-Stepped or Time-Stepped federates. If the federation is time managed, the logical time cannot run ahead of the rest of the federation, nor can the federation run off without it.

An event is any of the following that is associated with a particular point on the federation time axis:

- 1) A change of object attribute value
- 2) An interaction between objects
- 3) An instantiation of a new object
- 4) A deletion of an existing object

Object Management RTI services will include logical time information. Each federate will send events that include the logical time information. These events are: UPDATE ATTRIBUTE VALUES, SEND INTERACTION, and DELETE OBJECT INSTANCE. Each federate will receive events that include logical time information. These events are: REFLECT ATTRIBUTE VALUES, RECEIVE INTERACTION, and REMOVE OBJECT INSTANCE.

Only Time Stamped Ordered (TSO) events have temporal significance.

To ensure that events intended to be TSO are received as TSO, the federate should set its time switches before subscribing. To implement Time Management services correctly the following order will be used: Enable time constraint, enable time regulation, publish and subscribe, register initial instances, and then enter normal simulation loop.

The JIMM is an event-stepped model. Therefore, the federation will use NEXT EVENT REQUEST (NER) to request an advance of its clock. JIMM specifies with the request the logical time of the next event on its internal queue. The RTI responds in one of two ways: 1) The RTI calls back with TIME ADVANCE GRANT (TAG) and the time JIMM gave in the NEXT EVENT REQUEST. In this case, the RTI is guaranteeing that there is no event from the rest of the federation before the internal event. JIMM logical time then moves to the time of its next internal event. Therefore, JIMM is free to remove that event from its queue and process it, perhaps producing other internal or external events. 2) The RTI calls back with an external event (REFLECT ATTRIBUTE VALUES, RECEIVE INTERACTION, or REMOVE OBJECT INSTANCE) with a time stamp before the requested time, and then with TIME ADVANCE GRANT carrying the time of the external event. In this case, the RTI is guaranteeing that there will be no external event before the one just delivered. The federate logical time moves to the time of the event, and it is free to process that event, perhaps generating external or internal events.

## **2.2 Execute using time management executive in DSM mode**

In the DSM mode, execute with a separate time management executive which implements HLA-like time management architecture. This capability shall allow for the analysis and repeatability of J2C execution in a conventional configuration of assets linked to JIMM through shared memory interfaces. This separate time management executive will be a separate process from JIMM and the other assets but will be an asset. It will perform the functions listed in Section 2.0 above in addition to controlling the start of the exercise. The user interface will be very simple initially, enter a carriage return to start the exercise. If time and resources permit, the interface could be made more elaborate.

## **3. Planned JIMM Implementation**

The following sections describe components associated with the implementation of time management in JIMM. Figures 3.1-3 shows the basic sequence associated with this concept.

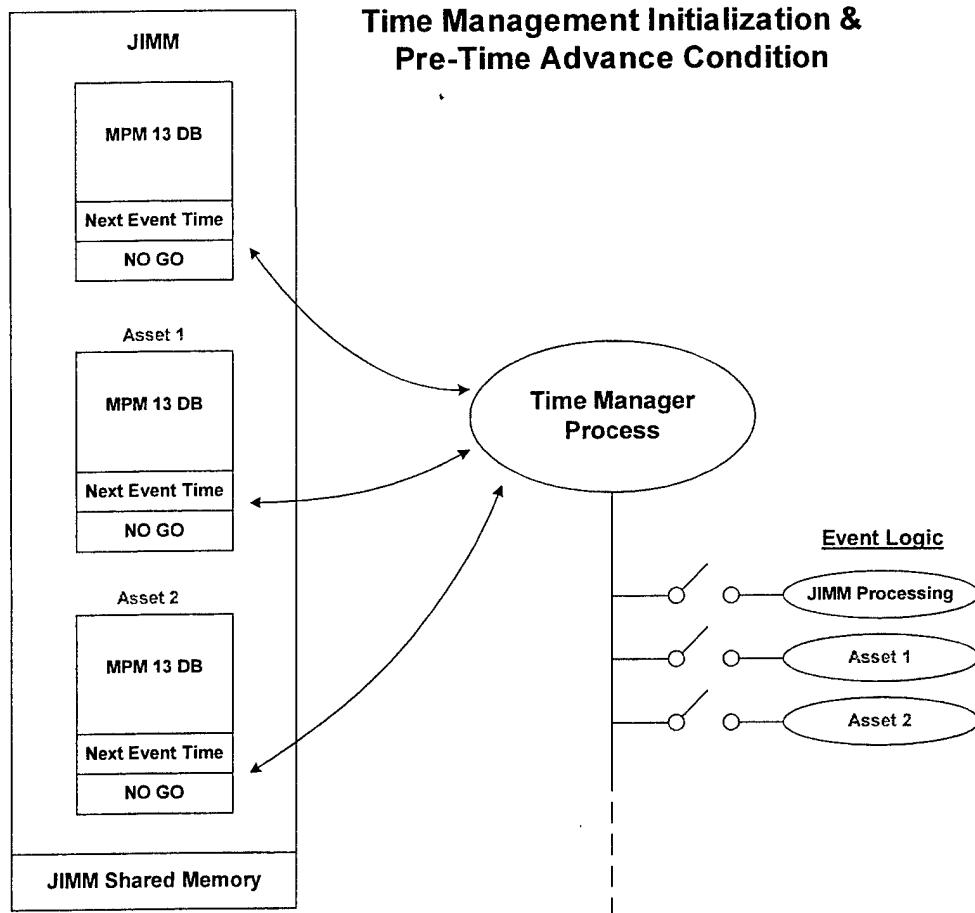


Figure 3.1. Time Management Initialization and Pre-Time Advance Condition.

In each asset header data block (MPM 13) in JIMM shared memory, an asset will store the time of the next event for that asset. In Figure 3.1, all of the assets have updated the time of their next event. For JIMM, that will be just the time of the next event. For other assets that might be the next event time, the minimum of all of the next event times, the next frame time, maximum game time (or MAX\_FLOAT) if no events are scheduled, etc. Except for the HLA interface asset, all assets will update their next event time in the MPM 13. The Time Manager Process, which in the DSM case is a separate process and in the HLA case is part of the HLA interface, polls all of the assets to determine the lowest next event time. There may be a tie with several assets. All assets will then wait for the “Go” flag to be set in their MPM 13.

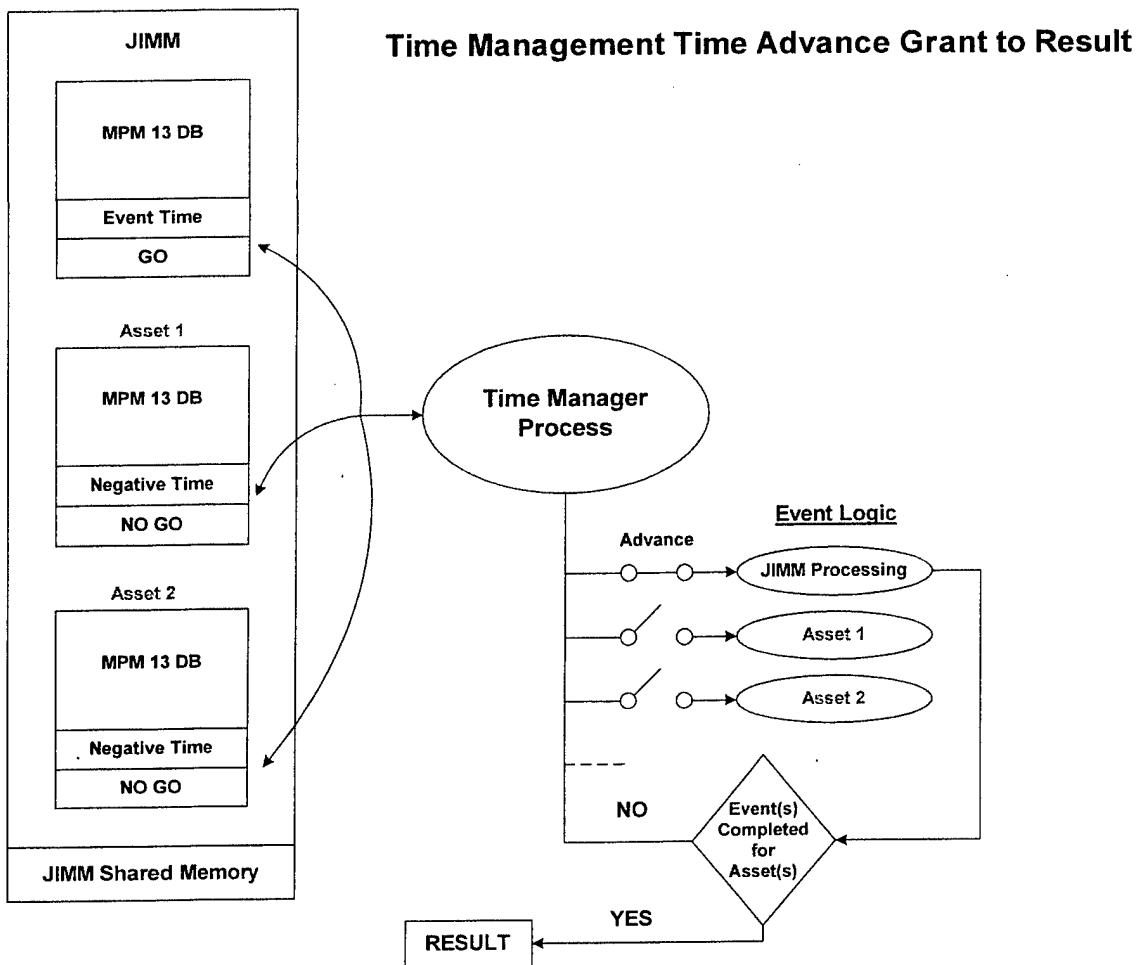


Figure 3.2. Time Management Time Advance Grant to Result.

Figure 3.2 shows the next step. Either the shared memory Time Manager or the HLA interface will set all next event times to a negative, set the “Go” flag in the appropriate asset(s) to 1 and set the “Go” in all of the rest of the assets to 0. The asset(s) that have their “Go” flag set will then process that event or time frame and will reset the flag to -1 when done.

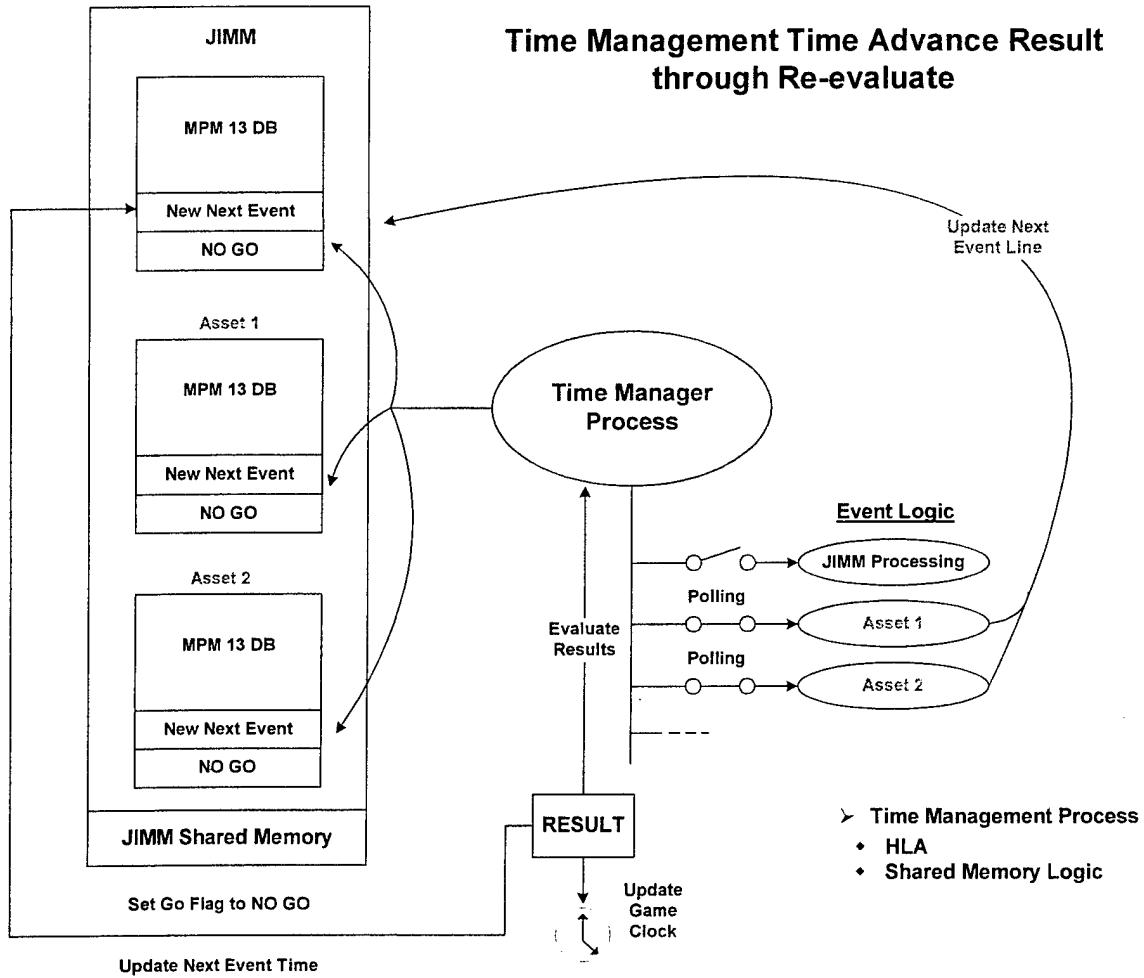


Figure 3.3. Time Management Time Advance Result through Re-evaluation.

Figure 3.3 shows the activity after the asset(s) with the “Go” flag set have completed their processing and reset the flag to -1. Then all assets will process the results of the Go and determine their new, possibly the same, next event time, and set it in the MPM 13 replacing the negative value that is there. Either the JIMM HLA interface or the Time Management process will be responsible for polling for the next event. The HLA interface will pass the event time to the RTI. In the case of an HLA federation, the RTI through its Time Management Services will grant a time advance that will come through the HLA interface; the HLA interface will then set the appropriate Go flag(s). In the case of DSM time management, the Time Management process will make this determination and set the appropriate Go flag(s).

### 3.1 Data Structures

During initialization a new dispatch (MPM 19), 119000, will be sent to every asset to inform them that time management procedures are in effect and the size of the time frame. The asset header data block (MPM 13) in JIMM shared memory will be enhanced to include two new elements, `next_event_time` and a `go_flag`.

## 3.2 JIMM Input

### 3.2.1 JIMM Input Syntax

The format for the time management input in the CDB file under the INTEGRATED-OPERATIONS data item is as follows:

```
USE-TIME-MANAGEMENT: TIME-FRAME-SIZE <elapsed-time.> <time-UOM> a/o  
LOOK-AHEAD <elapsed-time.> <time-UOM> a/o  
TIME-CONSTRAINED a/o TIME-REGULATED
```

An elapsed-time value less than or equal to 0.0 disables the time frame feature and causes the time management to be event based.

Figure 3.4 shows the frame-based concept. In the frame-based system everyone steps forward a frame time increment at a time. This system has the following assumptions:

### Time Management Viewpoints

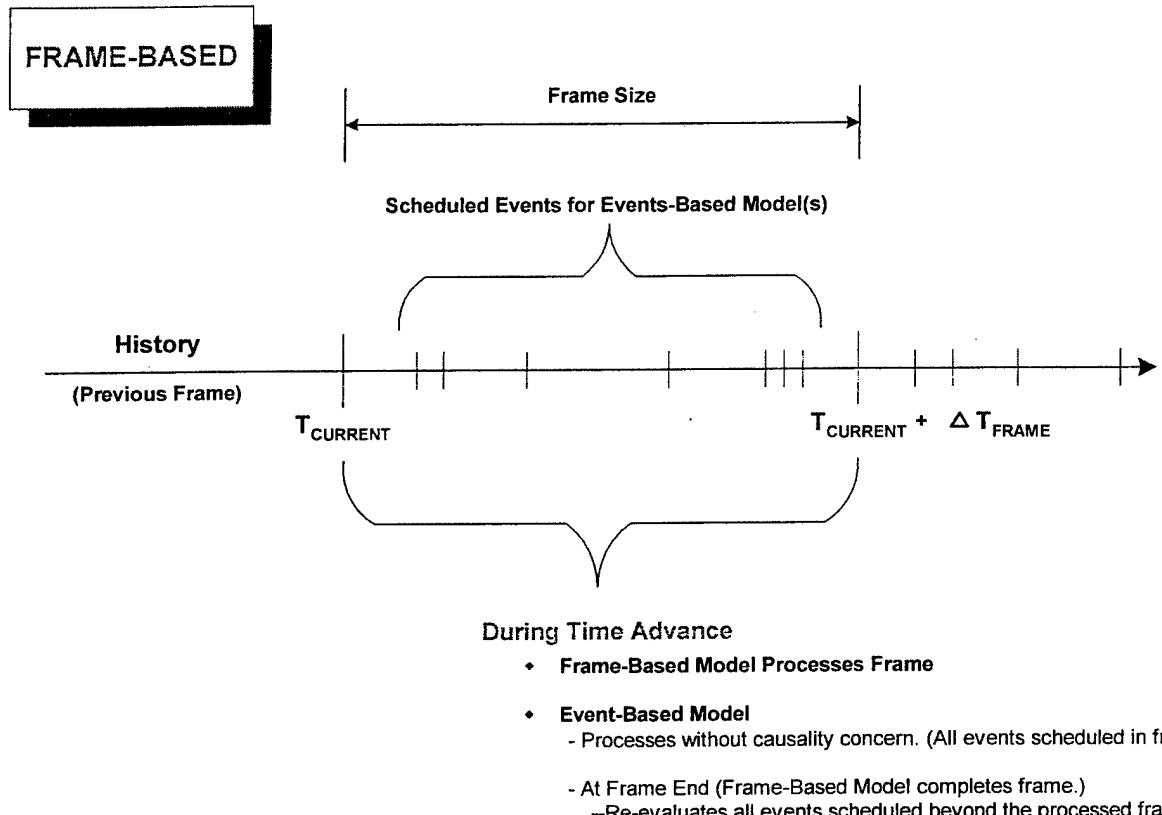


Figure 3.4. Frame-based, Time Management Concept.

1. Anything that happens at the beginning of the frame in one asset will not cause events to occur in another asset before the end of the frame (or can be made that way).
2. All assets will publish when they are ready to move to the next frame.
3. Frame size will be a user input and will have to be tuned to the assets involved.

4. This approach naturally integrates with frame-based simulations/simulators/stimulators.

Figure 3.5 shows the event-based concept. In the event-based system (where the TIME-FRAME-SIZE is less than or equal to 0) each asset posts the time of its next event. Some asset receives a time advance grant. After the time advance is processed, all assets re-evaluate the next event time based on the results of that activity. This system has the following assumptions:

### Time Management Viewpoints

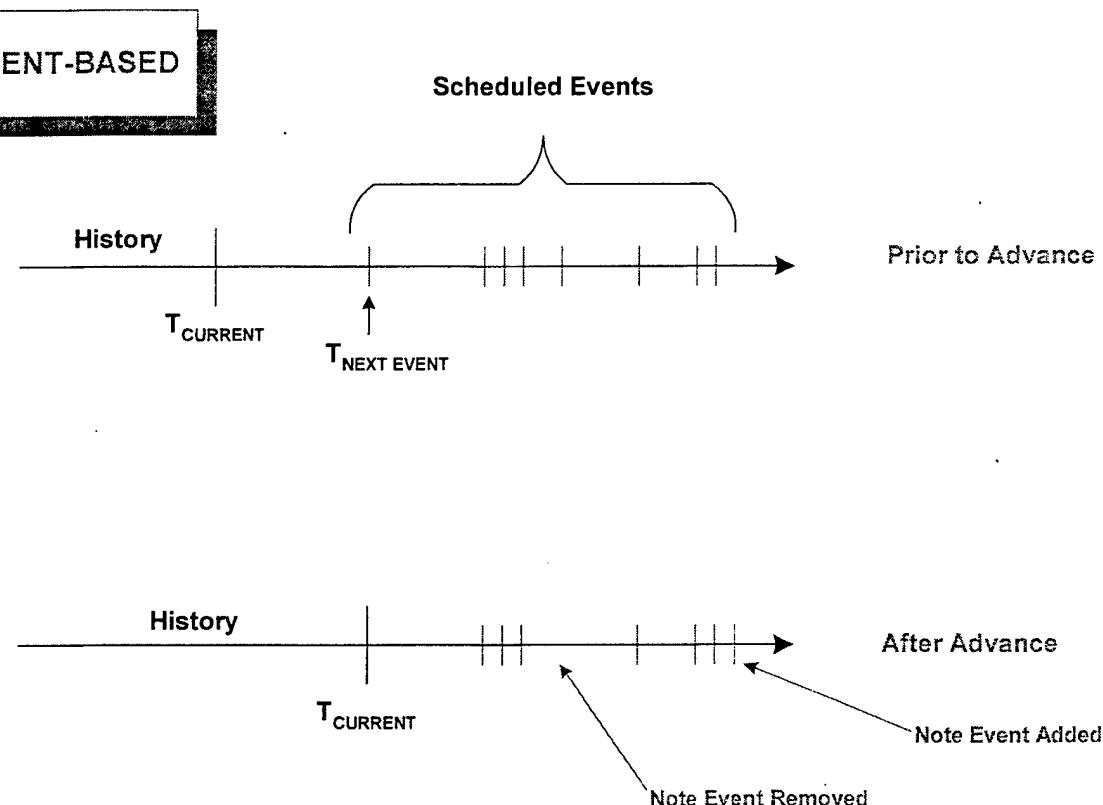


Figure 3.5. Event-based Time Management Concept.

1. This system will be slower in that each asset will have to re-evaluate next event as a result of possibly hidden results (a neighbor event).
2. All assets will have to publish the time of their next event.
3. All events at the same current time will be evaluated at once.
4. This system will probably result in moving ahead in very small time steps at times.

There are two additional optional data items, TIME-CONSTRAINED and TIME-REGULATED that can appear as well. The presence of TIME-CONSTRAINED means the federate is time constrained, TIME-REGULATED means the federate is time regulated. The absence of one or both items means that the federate does not have the missing characteristic(s).

### ***3.2.2 Target Routines***

This section lists the functions (and filenames) to be modified in order to implement the new time management capability in JIMM. Nwrk.cpp will be enhanced to read in the new input, store it, and send 170000 dispatches to all of the assets.

## **3.3 Code Description**

The following subsections describe how time management will be implemented in JIMM code.

### ***3.3.1 Target Routines***

This section lists the functions (and filenames) to be modified in order to implement the new time management capability in JIMM. The main loop in simnxt.cpp will be modified to post either the time of the next event or the time of the next frame containing an event and wait for a time advance grant from either the RTI through the HLA interface or the Time Manager process. Nwrk.cpp will be enhanced to provide templates for and receive a second new dispatch, 170001, which is a response from the assets acknowledging the receipt of the time management dispatch. The exercise will not start until all assets have acknowledged and set their assets to ready.

### ***3.3.2 Algorithmic Changes***

In each asset header data block (MPM 13) in JIMM shared memory, an asset will store the time of the next event for that asset. For JIMM, that will be just the time of the next event. JIMM will then wait for the “Go” flag to be set in its MPM 13. If the JIMM “Go” flag is set, then it will process the next event and reset the flag to -1 when done. In any case it will poll the “Go” flags that are set and when none are set, it will process the results of the Go and determine a new, possibly the same, next event time, and set it in the MPM 13 replacing the negative value that is there.

## **3.4 Output Description**

There is no specific output for time management.

## **4. Testing Approach**

The following sections describe the testing of the new time management capability in JIMM. The testing will be broken into two parts, HLA Time Management services and direct, shared memory time management. The key issue is to maintain causality, so the testing will look for event times earlier than the current time. Hopefully there will be no such events. JIMM processing any events received via Data Interactions from the RTI with a time tag less than the current JIMM time indicates a test failure. Similarly, JIMM publishing any Data Interactions with a time tag less than the current TAG + look-ahead will also indicate a test failure.

### **4.1 HLA Time Management Services**

A test federation will be created with the following components:

- J2C (the JIMM to CART I/F) as an asset
- One or more CART federates, one for each HPM [for the purposes of testing, these may be replaced with a CART emulator if actual CART federates aren’t available]
- JIMM as the MASTER-MODEL asset

- JIMM's HLA I/F as an HLA-INTERFACE asset [Note: If J2C also serves as JIMM's HLA I/F, the HLA I/F asset is not needed]
- The HLA RTI ("fedex") and an RTI console
- The "dummy" federate

#### ***4.1.1 HLA Time Management, Baseline Case***

JIMM's CDB will specify TIME-CONSTRAINED and TIME-REGULATED, with no LOOK-AHEAD time. This case should result in JIMM being strictly time managed (i.e., both regulated and constrained) with a default look-ahead of 0.0001 seconds.

##### Federation Creation:

- Check that the "dummy" federate creates the federation and holds time until all federates are ready. Each federate/simulation's local time will be examined to ensure that time is not advancing.
- Check that JIMM's HLA interface invokes the RTI calls ENABLE TIME CONSTRAINED and ENABLE TIME REGULATION, and processes the RTI callbacks TIME CONSTRAINED ENABLED and TIME REGULATION ENABLED.
- Check that action requests and responses are processed appropriately.

##### Federation Execution:

- Once time starts advancing, check that J2C sends Data Interactions to the RTI with the latest TAG + LOOK-AHEAD.
- Check that when JIMM's next\_event\_time is  $\leq$  TAG, that incoming Data Interactions are processed and that JIMM periodically ticks the federation.
- Check that when JIMM's next\_event\_time is  $>$  TAG, that J2C makes a NER request.
- Since JIMM is time constrained, ensure that JIMM doesn't process any internal events beyond its current LBTS.
- Ensure that JIMM doesn't advance time until a TAG is received.

#### ***4.1.2 HLA Time Management, Look-Ahead Input Case***

Repeat case 4.1.1 with LOOK-AHEAD 0.0001 (SEC) in the CDB. The results should be identical to 4.1.1.

#### ***4.1.3 HLA Time Management, Different Look-Ahead Case***

Repeat case 4.1.1 with LOOK-AHEAD 0.0002 (SEC) in the CDB. Check that JIMM and J2C are using the larger look-ahead value to process events and time advances.

#### **4.1.4 HLA Time Management, Time Regulated Case**

Repeat case 4.1.1 with TIME-CONSTRAINED removed from the CDB. [Note: This case assumes that the JIMM scenario and CART server emulator can be made to run so that JIMM is expected to publish Data Interactions but not receive any.] Check that JIMM is regulating and CART is constrained, but not vice versa.

#### **4.1.5 HLA Time Management, Time Constrained Case**

Repeat case 4.1.1 with TIME-REGULATED removed from the CDB. [Note: This case assumes that the JIMM scenario and CART server emulator can be made to run so that JIMM is expected to receive Data Interactions but not publish any.] Check that CART is regulating and JIMM is constrained, but not vice versa.

#### **4.1.6 HLA Time Management, Frame-Based Case**

Repeat case 4.1.1 with TIME-FRAME-SIZE 0.02 (SEC) specified in the CDB. Check that JIMM and J2C are exhibiting the same characteristics as 4.1.1, but running with 50 Hz frames (rather than event-based).

### **4.2 DSM Time Management Services**

An integration testbed will be created with the following components:

- J2C (the JIMM to CART I/F) as an asset
- One or more CART assets, one for each HPM [for the purposes of testing, these may be replaced with the CART emulator if actual CART assets aren't available, *and* the CART server emulator is set up to work in DSM mode]
- JIMM as the MASTER-MODEL asset
- TME (Time Management Executive) as a separate process (but not an asset)

#### **4.2.1 DSM Time Management, Baseline Case**

JIMM's CDB will specify TIME-CONSTRAINED and TIME-REGULATED, with no LOOK-AHEAD time. This case should result in JIMM being strictly time managed (i.e., both regulated and constrained) with a default look-ahead of 0.0001 seconds.

##### Scenario Creation:

- Check that all assets acknowledge the time management dispatch from the TME
- Check that action requests and responses are processed appropriately by the TME

##### Federation Execution:

- Once time starts advancing, check that J2C sends Data Interactions to the TME with the latest TAG + LOOK-AHEAD

- Check that while JIMM's next\_event\_time is  $\leq$  TAG, that incoming Data Interactions are processed and that JIMM periodically ticks the federation
- Check that while JIMM's next\_event\_time is  $>$  TAG, that J2C makes a NER request
- Since JIMM is time constrained, ensure that JIMM doesn't process any internal events beyond its current LBTS
- Ensure that JIMM doesn't advance time until a TAG is received
- Check that TME is managing time per figures 3.1 – 3.3

#### ***4.2.2 – 4.2.6 DSM Time Management, Other Cases***

Repeat case 4.1.2 – 4.1.6, using the TME instead of the HLA RTI. Compare scenario execution from case 4.1.X with case 4.2.X to see if the same event streams are generated.

## Appendix B: Interface Control Document for Data Interactions between JIMM and the CART Middleware

The communication protocol between CART and JIMM was established by defining the data interactions that would take place between the CART Middleware and JIMM. This Interface Control Document defines the structure and content of those data interactions. The interactions consist of two subscription interactions and five data interactions:

- Subscription interactions
  1. Subscription Services Data – data used to subscribe entities to specific HPM services
  2. Initialization Data – initialization data for each entity subscribing for HPM service
- Data interactions
  1. Track Data – contains track information to be communicated between entities
  2. Track Assignment – contains track assignment information
  3. Cancel Track Assignment – contains track cancel information
  4. HF Request – used by an entity to request track height information
  5. Engagement Status – contains track engagement status information

Table B-1 contains the Subscription Services Data - Data used to subscribe entities to specific HPM services. This data will be mapped to "Receive As Event" and will cause the HPM to start appropriate values of "tag" for each entity subscribing to an HPM service. Every data interaction sent following subscription will contain a Fixed Datum ID of 300000 and a Fixed Datum Value containing the Service Subscription number.

**Table B-1. Subscription Services Data.**

	Type	Data Enumeration	Header Info	Comments
SOC Service Subscription	int32	44		Receipt of this data will trigger an entity "tag" to start in the SOC part of the HPM.
CRC Service Subscription	int32	45		Receipt of this data will trigger an entity "tag" to start in the CRC part of the HPM.
CRP Service Subscription	int32	46		Receipt of this data will trigger an entity "tag" to start in the CRP part of the HPM.

Table B-2 shows initialization data that each entity uses for subscribing to a HPM service. All data will be sent in the second action request during the subscription process.

**Table B-2. Initialization Data.**

	Type	Data Enumeration	Header Info	Comments
Entity ID	int32	32		ID of the subscribing entity.
Entity Type	int32	30		Entity type for the subscribing entity. A value of 1 indicates a SOC, 2 indicates a CRC, and 3 indicates a CRP.
Entity X Position	float 64	27		X position of the subscribing entity from scenario center in meters. East is positive X.
Entity Y Position	float 64	28		Y position of the subscribing entity from scenario center in meters. North is positive Y.
Cultural Parameter: Familiarity with Enemy	int32	33		The extent to which a culture has had prior experience with an enemy. A value of 2 indicates a high familiarity with the enemy, and 1 indicates low familiarity with the enemy.
Cultural Parameter: Willingness to Take Risks	int32	34		A culture's willingness to make difficult decisions in vulnerable situations risking the consequences of errors. A value of 2 indicates a high willingness to take risk, and 1 indicates low willingness to take risk.
Cultural Parameter: Distribution of Power	int32	35		The perceived difference in power between an individual and his superior. A value of 2 indicates a high distribution of power, and 1 indicates low distribution of power.
Number of Subordinates	int32	36		Number of subordinate entities reporting to the subscribing entity. (CRCs to a SOC, CRPs to a CRC, and Radars to a CRP).
Subordinate Entity ID	int32 array [8]	37		ID for each subordinate entity reporting to the subscribing entity. Each entity may have a maximum of eight subordinates.
Subordinate Entity Type	int32 array [8]	43		The type of subordinate reporting to the subscribing entity. A value of 1 = HF Radar, 2 = 2D Radar, 3 = 3D Radar, 4 = CRP, 5 = CRC, and 6 = SAM.

Subordinate X Position	float 64 array [8]	38		X position for each subordinate entity reporting to the subscribing entity.
Subordinate Y Position	float 64 array [8]	39		Y position for each subordinate entity reporting to the subscribing entity.
Scenario Type	int32	54		A value of 1 = PreWar, 2 = Traditional, and 3 = SAMbush mode.
IADS Mode of Operations	int32	55		A value of 1 = Manual Operations, and 2 = Automatic Operations.

Table B-3 through Table B-5 shows track data. All of this data that is not header information will be indexed by tag in the HPM assuring that the data goes into the correct HPM service.

**Table B-3. Track Data.**

	Type	Data Enumeration	Header Info	Comments
Message Type = 1006	int32	none		JIMM message definition number.
Send Time	float 64	none	HLA Header	Message header info. Federate simulation time stamp in seconds.
Perception Time	float 64	40		Time in which the track is originally perceived. Simulation time in seconds.
Track Data New flag	int32	26		A flag indicating that track data has been sent from JIMM to CART.
Track Data Sent flag	int32	48		A forced value sends "packaged" track data from CART to JIMM.
Sender Entity ID	int32	20		Identifies the subordinate sender of the track information.
Receiving Entity ID	int32	41		Used by JIMM to ascertain where the track data goes after receiving a message from CART.
Track Entity ID	proto col dependent	none	HLA Header	Identifies specific track for which data is being transmitted.
Track X Position	int32	24		X position of the track from scenario center in meters. East is positive X.
Track Y Position	int32	none		Y position of the track from scenario center in meters. North is positive Y.
Track Heading	float 64	15		Degrees clockwise from true North.
Track Speed	float 64	16		Meters per second.
Track Altitude	float 64	13		In meters. Negative if unavailable.
Aircraft Type	float 64	14		Ground truth categorization/classification of aircraft. 1 = Bomber, 2 = Fighter, 3 = Low Observable, 4 = Airborne Command, and 5 = Other or UAV.
IFF Enumeration	float 64	17		1 = Friend, 2 = Hostile, 3 = Unknown, and 4 = Civilian.

**Table B-4. Track Assignment Data.**

	Type	Data Enumeration	Header Info	Comments
Message Type = 2002	int32	none		JIMM message definition number.
Send Time	float 64	none	HLA Header	Message header info. Federate simulation time stamp in seconds.
Sender Entity ID	int32	20		Identifies the sender of the track assignment information.
Receiving Entity ID	int32	42		Used by JIMM to ascertain where the track assignment data goes after receiving a message from CART.
Track Assignment Sent flag	int32	57		A forced value sends "packaged" track assignment data from CART to JIMM.
Track Entity ID	int32	21		The track entity ID for the track being assigned.
Track X Position	int32	15		X position of the track from scenario center in meters. East is positive X.
Track Y Position	int32	16		Y position of the track from scenario center in meters. North is positive Y.
Track Heading	float 64	13		Degrees clockwise from true North.
Track Speed	float 64	14		Meters per second.
Track Altitude	float 64	17		In meters. Negative if unavailable.
Aircraft Type	float 64	2		Ground truth categorization/classification of aircraft. 1 = Bomber, 2 = Fighter, 3 = Low Observable, 4 = Airborne Commander, and 5 = Other or UAV.
IFF Enumeration	float 64	11		1 = Friend, 2 = Hostile 3 = Unknown, and 4 = Civilian.
Fire Authorization Type Flag	int32	10		Indicates the type of track assignment ordered by the SOC. 0 = normal assignment without fire authorization, 1 = normal track assignment with a fire authorization, 2 = SAMbush assignment without fire authorization, 3 = SAMbush assignment with fire authorization.

**Table B-5. Cancel Track Assignment Data.**

	Type	Data Enumeration	Header Info	Comments
Message Type = 2004	int32	none		JIMM message definition number.
Send Time	Float 64	none	HLA Header	Message header info. Federate simulation time stamp in seconds.
Sender Entity ID	int32	20		Identifies the sender of the cancel track assignment information.
Receiving Entity ID	int32	42		Used by JIMM to ascertain where the track cancellation data goes after receiving a message from CART.
Track Cancellation Sent flag	int32	58		A forced value sends "packaged" track cancellation data from CART to JIMM.
Track Entity ID	int32	22		The track entity ID for the track being cancelled.

Table B-6 shows the height finding request data. When the CRP receives track data that does not include height information and it is needed by the CRP before sending on to the CRC, it will use this request.

**Table B-6. Height Finding Request Data.**

	Type	Data Enumeration	Header Info	Comments
Message Type = 2005	int32	none		JIMM message definition number.
Send Time	float64	none	HLA Header	Message header info. Federate simulation time stamp in seconds.
Sender Entity ID	int32	20		Identifies the sender of the HF request.
Receiving Entity ID	int32	42		Used by JIMM to ascertain where the HF request goes after receiving a message from CART.
HF Request Sent flag	int32	56		A forced value sends "packaged" HF Request data from CART to JIMM.
Track Entity ID	int32	18		The track entity ID for the HF request track.

Table B-7 shows the engagement status data.

**Table B-7. Engagement Status Data.**

	Type	Data Enumeration	Header Info	Comments
Message Type = 3016	int32	none		JIMM message definition number.
Send Time	float64	none	HLA Header	Message header info. Federate simulation time stamp in seconds.
Sender Entity ID	int32	20		Identifies the sender of the engagement status report.
Track Engagement Status Flag	int32	52		A forced value indicates that a track engagement status message has been sent.
Track Entity ID	int32	53		The track entity ID for the engagement status report.
Engagement Status	int32	9		1 = Fired on and destroyed, and 2 = fired on and missed.

## **Appendix C: CART Human Performance Model for IADS Scenario**

The following report documents the Integrated Air Defense System Model created for the CMC2 project.

**Scientific and Technical Report  
for  
Cultural Modeling of Command and Control**

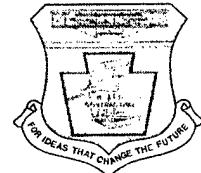
**Integrated Air Defense System Model**

*Prepared for:*

*Air Force Research Laboratory (AFRL/HESS)*

2698 G Street, Bldg 190

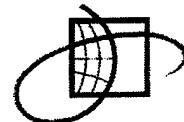
Wright-Patterson AFB, OH 45433-7604



*Prepared by:*

**Micro Analysis & Design, Inc.**

4949 Pearl East Circle, Suite 300  
Boulder, CO 80301-2477



Micro Analysis & Design

## TABLE OF CONTENTS

<b>1.0 PAPER OVERVIEW .....</b>	<b>68</b>
1.1 ARCHITECTURE.....	68
<b>2.0 IADS MODEL .....</b>	<b>69</b>
2.1 SECTOR OPERATIONS CENTER (SOC).....	70
2.1.1 <i>Monitoring Air Sector</i> .....	71
2.1.2 <i>Threat Evaluation</i> .....	71
2.1.3 <i>Track Prioritization</i> .....	75
2.1.4 <i>Track Assignment</i> .....	75
2.1.5 <i>Receive Engagement Status Reports</i> .....	83
2.2 CONTROL AND REPORTING CENTER (CRC) .....	83
2.2.1 <i>Monitoring Track Data</i> .....	84
2.2.2 <i>Track Evaluation</i> .....	85
2.2.3 <i>Send Track Data</i> .....	85
2.2.4 <i>Delete Stale Track Data</i> .....	85
2.3 CONTROL AND REPORTING POST (CRP) .....	85
2.3.1 <i>Monitoring Track Data</i> .....	86
2.3.2 <i>Track Evaluation</i> .....	86
2.3.3 <i>Delete Stale Track Data</i> .....	86
<b>3.0 DATA MESSAGES.....</b>	<b>86</b>
<b>4.0 DEVELOPING LIMITATIONS.....</b>	<b>90</b>

## List of Figures

FIGURE 1-1. CLIENT-SERVER ARCHITECTURE WITH JIMM AND CART .....	68
FIGURE 1-2. GENERIC IADS COMMAND AND CONTROL LAYOUT.....	69
FIGURE 2-1. IADS TOP LEVEL NETWORK.....	70
FIGURE 2-2. SOC OPERATOR TASKS .....	71
FIGURE 2-3. IADS AIR ZONES.....	76
FIGURE 2-4. CRC TASK NETWORK. ....	84
FIGURE 2-5. CRP TASK NETWORK.....	85

## List of Tables

TABLE 2-1. EVALUATION SUMMARY.....	72
TABLE 2-2. EVALUATION CRITERIA. ....	73
TABLE 2-3. WEIGHTING FACTORS.....	74
TABLE 2-4. (X, Y) POSITION EVALUATION CRITERIA.....	74
TABLE 2-5. AIR SPEED EVALUATION CRITERIA.....	74
TABLE 2-6. ALTITUDE EVALUATION CRITERIA .....	75
TABLE 2-7. HEADING EVALUATION CRITERIA. ....	75
TABLE 2-8. TRACK TYPE EVALUATION CRITERIA. ....	75
TABLE 3-1. INITIALIZATION DATA .....	87
TABLE 3-2. TRACK DATA MESSAGE.....	88
TABLE 3-3. TRACK ASSIGNMENT DATA MESSAGE.....	89
TABLE 3-4. TRACK CANCELLATION DATA MESSAGE.....	89
TABLE 3-5. ENGAGEMENT STATUS REPORT MESSAGE.....	90
TABLE 3-6. HEIGHT FINDING REQUEST DATA.....	90

## List of Flowcharts

FLOWCHART 2-1. PREWAR TRACK ASSIGNMENT. ....	78
FLOWCHART 2-2. TRADITIONAL TRACK ASSIGNMENT. ....	80
FLOWCHART 2-3. SAMBUSH TRACK ASSIGNMENT. ....	82
FLOWCHART 2-4. RECEIVE ENGAGEMENT STATUS REPORT. ....	83

## 1.0 Paper Overview

The team of Micro Analysis and Design (MA&D) and Northrop Grumman have created a simulation model of an Integrated Air Defense System (IADS) to include Command and Control (C2) operators and how their human behavior is affected by cultural differences. This paper will discuss the IADS model and associated cultural differences for Iraq and North Korea operations. MA&D and Northrop Grumman demonstrate cultural behavior differences by providing high fidelity models of cultural traits and military strategies. This collaboration yielded a model that provides a realistic representation of an IADS and its C2 node operators.

### 1.1 Architecture

System architecture included two existing simulation environments: the Joint Interim Mission Model (JIMM), a general purpose, data-driven, conflict simulation generator; and the Combat Automations Requirement Testbed (CART), a high fidelity human performance modeling application. JIMM and CART interoperate via a “client-server” architecture allowing entities in JIMM to be served by high fidelity human performance models in CART. Interoperability of the two simulations can be made using either the High Level Architecture (HLA) or high speed Direct Shared Memory (DSM) using a Run Time Infrastructure (RTI). (See Figure 1-1)

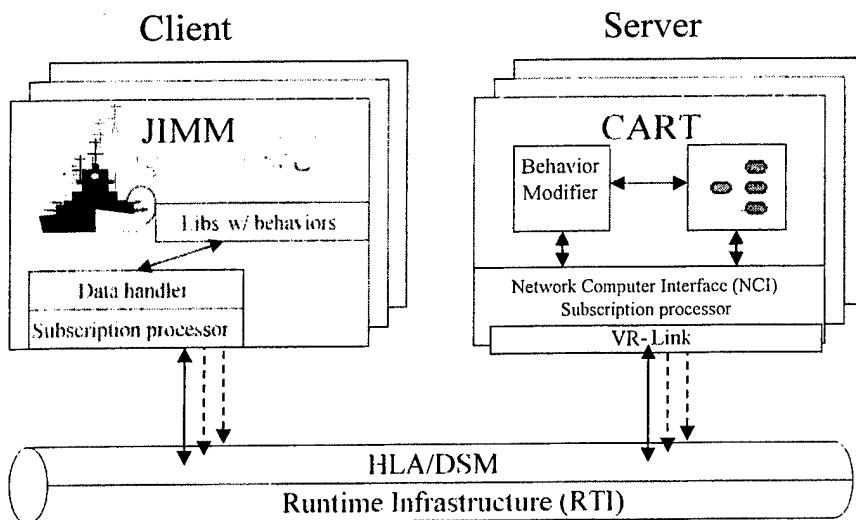


Figure 1-1. Client-server architecture with JIMM and CART.

JIMM uses a subscription process to link entities with human performance models in CART. This subscription maintains a relationship between the simulations as well as providing load balancing between multiple clients and multiple servers and the human behavior model services they provide. That is, more than one entity can be simulated within the same model simultaneously. With this capability JIMM can subscribe to a single human performance model (HPM) for many IADS nodes within the same simulation exercise. The efficiency gained is a minimum number of applications within a federation to perform all human performance modeling. One can also configure CART to run on numerous processors as federates in a simulation to provide higher performance. That is, CART and the IADS models are scalable and flexible.

To represent the actions and decisions made by IADS personnel MA&D created a task network model within CART depicting three critical nodes of a generic IADS: 1) A Sector Operations Center (SOC), the highest level of the command chain where key decisions regarding missile fly outs occur, 2) A Control and Reporting Center (CRC), the intermediary between the SOC, the Surface to Air Missile Operations (SAM Ops) sites, and the radar Ops, and 3) A Control and Reporting Post (CRP), the radar Ops where intelligence operators detect aircraft within the air sector. Figure 1-2 shows the IADS architecture MA&D used for the integrated simulation (i.e. the combined JIMM and CART effort). The layout is a generic representation of an IADS and is not intended to replicate any specific country's IADS. Each rectangle in the figure represents an entity within an IADS that exists in JIMM and can be represented by a HPM in CART. For this effort, only the three shaded rectangles (a SOC, CRC, and CRP) are entities modeled in CART.

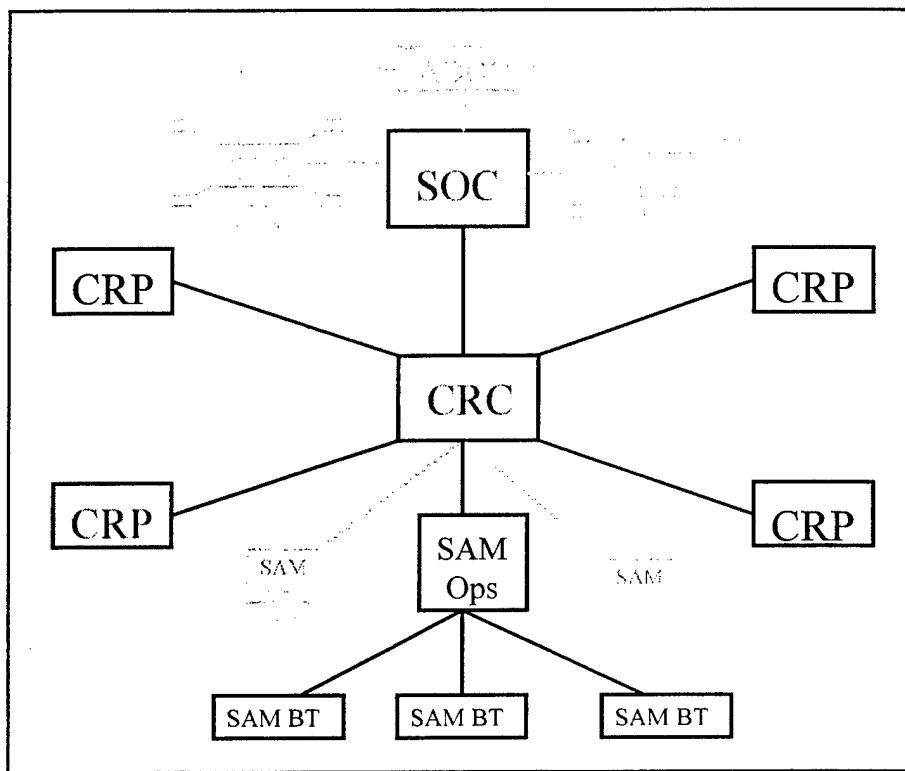


Figure 1-2. Generic IADS Command and Control layout.

MA&D designed task networks, for each of the three modeled IADS nodes, to reflect the logic and actions of the respective IADS personnel. To encapsulate these tasks, MA&D created three separate network models, one for each node (named SOC, CRC, and CRP). Currently, only the SOC, described in section 2.1, takes into account human behavioral variations caused by cultural differences.

## 2.0 IADS model

This section describes the CART IADS model in detail. Figure 2-1 displays an overview of the IADS HPM. The top row of nodes (ovals), numbered 7, 20, 21, 26, and 62, are federation

initialization and management processes for interoperability of CART with JIMM. The remaining left-hand column of nodes, numbers 22, 23, 24, and 25, contain databases for holding messages communicated between the two simulations. (Section 3.0, "Data Messages," contains a list of all data messages communicated between JIMM and CART). The three rectangles in the figure, numbered 1, 2, and 3, contain networks of nodes that model the IADS nodes and human behaviors of interest within them.

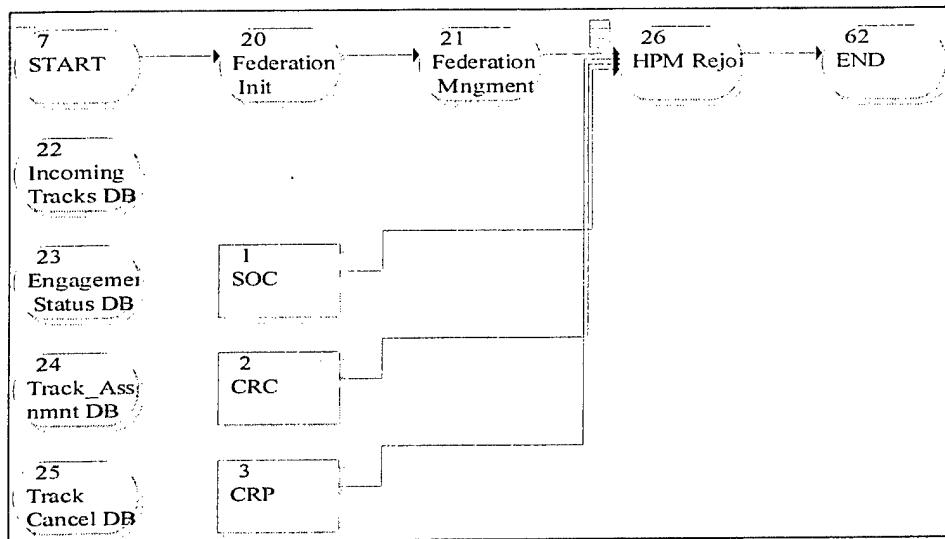


Figure 2-1. IADS top level network.

Subsequent sections discuss each node in detail, beginning with the SOC in sub-section 2.1, continuing with the CRC in sub-section 2.2, and concluding with the CRP in sub-section 2.3.

## 2.1 Sector Operations Center (SOC)

The SOC operator is the highest level of command where critical decisions are made concerning the defense of a large volume of airspace against invading aircraft. At the SOC's disposal is an integrated network of surface to air missiles (SAMs) that it can call upon to counter an invading attack. Figure 2-2 shows the task network model of the command and control functionality at the SOC.

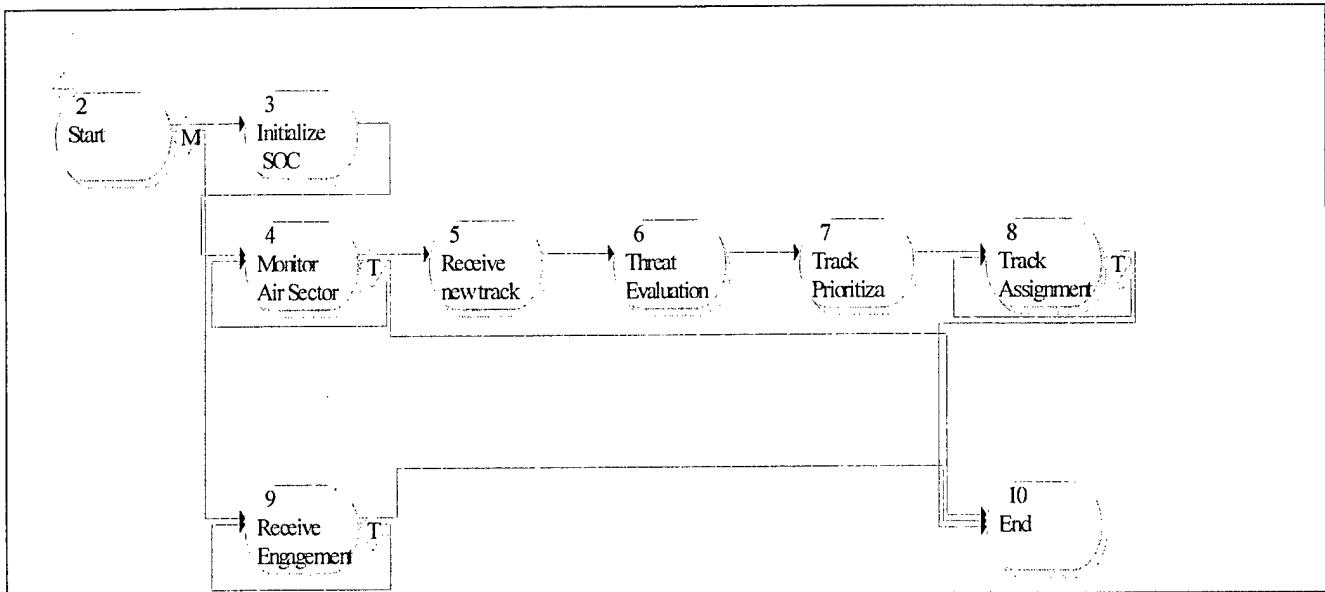


Figure 2-2. SOC operator tasks.

When a JIMM entity subscribes for SOC HPM services, task 2, the “Start” task, commences and simultaneously starts two continuous task paths, one beginning with task 3, “Initialize SOC,” and another beginning with task 9, “Receive Engagement Status Reports.” Task 3 initializes the threat evaluation process, discussed in more detail in sub-section 2.1.2, and starts a repeating set of tasks starting with task 4, “Monitor Air Sector” and continuing to task 8, “Track Assignment.” The other continuous section is the looping of task 9, “Receive Engagement Status Reports.”

This HPM abstracts complex behaviors performed by the SOC into five essential functions: 1) Air Sector Monitoring, 2) Threat Evaluation, 3) Track Prioritization, 4) Track Assignment, and 5) Receive Engagement Status Reports. (Note: A track in an IADS is an aircraft detected by the radar operator). The sub-sections below describe each function in detail.

### 2.1.1 Air Sector Monitoring

The SOC continuously monitors its air sector to determine if any aircraft within are threatening enough to justify immediate defensive action. Two and three dimensional radar systems gather track data about the air sector that propagates up to the SOC from the CRC via the CRP. As the track data supplied to the SOC cumulates, it forms a “picture” of the air sector revealing aircraft locations and other details such as height and speed. When the SOC receives new track data, updating its “picture” of the air sector, the SOC performs Threat Evaluation. If the SOC is already processing track information (in tasks 6 through 8) when new track data arrives, it will immediately continue to threat evaluation when it returns to task 4.

To replicate the SOC action of monitoring the air sector, task 4 is repeatedly performed. When the SOC receives new track data, monitoring ceases and task flow continues to task 5, “Receive New Track Data,” preparing the model for Threat Evaluation by deciphering new track data from existing track data. The next sub-section discusses threat evaluation in detail.

### 2.1.2 Threat Evaluation

Threat evaluation involves the SOC assessing the potential lethality of the newly updated tracks. The SOC assesses potential lethality based on evaluation parameters, evaluation criteria, and weighting factors.

*Evaluation parameters* describe characteristics of a track; such as air speed, altitude, and heading. These parameters help the SOC determine the lethality of a track. For example, an airplane traveling at an extremely high speed could indicate that an invading enemy is preparing for missile release on the IADS (traveling at high speed would allow for a quick efficient attack and an escape from SAMs released by the IADS). The high speed of the airplane provides an indication of missile release and therefore a threat to the IADS.

*Evaluation criteria* further define the potential lethality of each evaluation parameter. Criteria are given a value ranging from zero to ten. For example, consider the evaluation parameter air speed. The evaluation criteria for this parameter define the lethality for air speed for different speed ranges. Since high speeds can indicate invading threats preparing for missile strike, high speeds might be rated an eight on a scale of zero to ten in increasing lethality. Similarly, low speeds may indicate invading threats preparing for bombing, a very serious threat to a defense, and consequently rated a ten. Moderate speeds might receive a lower rating, as they are not particularly indicative of any type of attack.

*Weighting factors* set up the relative importance of each evaluation parameter. All weighting factors for the evaluation parameters must sum to 100. For example, if one country believed that air speed indicated a stronger threat than altitude and heading, then it would give air speed a heavier weighting factor than it would to altitude and heading (e.g. Air Speed Weighting Factor 50, Altitude Weighting Factor 25, and Heading Weighting Factor 25).

A threat evaluation algorithm first calculates a partial threat score for each evaluation parameter by multiplying the evaluation criteria rate by the weighting factor. The algorithm then aggregates the partial threat scores for each track into a total threat score. The maximum threat score is 1000. Higher scores indicate a more threatening situation. Evaluation summary (Table 2-1) and Evaluation criteria (Table 2-2) illustrate the essential features of the threat evaluation task for three hypothetical tracks, identified as Track 1, Track 2, and Track 3 (Evaluation criteria is shown for air speed only). Track 2 is the most threatening with a total threat score of 925.

**Table 2-1. Evaluation summary.**

Item	Evaluation Parameter	Weighting Factor	Track 1		Track 2		Track 3	
			Rate	Score	Rate	Score	Rate	Score
1	Air Speed	50	10	500	10	500	7	350
2	Altitude	25	7	175	10	250	7	175
3	Heading	25	8	200	7	175	8	200
<b>Total Threat Score</b>		<b>100</b>		<b>875</b>		<b>925</b>		<b>725</b>

**Table 2-2. Evaluation criteria.**

Rating (points)	Evaluation Criteria---Item 1: Air Speed
10	The track is traveling at speeds over 400 m/s.
8	The track is traveling at speeds between 250 m/s and 400 m/s.
7	The track is traveling at speeds between 150 m/s and 250 m/s.
10	The track is traveling at speeds below 150 m/s.

Human behavioral variations caused by cultural differences can be modeled by varying any of the aforementioned fields where cultural preferences might be different. Adjusting for differences can include varying the *ranges* in the evaluation criteria (e.g. tracks exceeding an air speed of 550 m/s instead of 400 m/s could be rated 10), varying the *rating factors* (e.g. tracks with an air speed over 400 m/s could be rated 6 instead of 10) and varying the *weighting factors* (e.g. air speed could be rated 15 and heading could be rated 60, instead of 40 and 25 respectively). Altering the evaluation criteria, the rating factors, or the weighting factors will lead to different threat scores and ultimately different critical decisions of which tracks to engage and in what order to engage them. Varying these fields gives the user flexibility to include cultural differences.

Table 2-3 through Table 2-8 present the variations MA&D chose for evaluation parameters, evaluation criteria, and weighting factors. **The values MA&D selected for the threat evaluation process are purely notional and intended to serve as a working example of how a country or culture might behave. These notional values reflect general descriptions of the two cultures from SME and literature sources for the sole purpose of demonstrating the functionality of the CMC2 tool and are not based on any actual data.**

The values for the evaluation criteria, weighting factors, and nearly all of the rating factors are the same for the two cultures (North Korea and Iraq), with the one exception being the evaluation criteria for (X, Y) position.

MA&D gave higher rates to closer proximity ranges for North Korea and lower rates to closer proximity for Iraq because of each cultures willingness to take risk (WR), one of the three cultural parameters created in this effort. Detailed information on each cultural parameter can be found in the final report for this effort, entitled “Scientific and Technical Report for Cultural Modeling of Command and Control Echelons.” WR has been defined as an individual's willingness to make decisions using incomplete information that place him in a vulnerable situation increasing the consequences of his errors. Military history and consultation with several subject matter experts (SMEs) regarding the behaviors of North Koreans and Iraqis lead to the assignment of a high WR for Korea and a low WR for Iraq. The United States past history of military engagements with North Korea reveals that it is a very structured and disciplined country whose combatants follow orders closely and intrepidly serve their country. Iraqis, conversely, have demonstrated opposite behaviors. At the beginning of the first Gulf War, Iraqi SAM operators behaved as expected, but as the war continued they soon learned that turning on their radar or firing a missile was tantamount to suicide as their positions were immediately compromised to U.S. intelligence. As a result, the Iraqi IADS system became ineffective because SAM operators would inefficiently fire at long ranges with low probability of kill (pK) levels to avoid becoming a U.S. target.

**Table 2-3. Weighting factors.**

	Evaluation Parameter	Weighting Factor
1	(X, Y) Position	60
2	Air Speed	15
3	Altitude	10
4	Heading	10
5	Track Type	5
	Total	100

**Table 2-4. (X, Y) Position evaluation criteria.**

Iraqi Rating (points)	North Korean Rating (points)	Evaluation Criteria---Item 1: (X, Y) Position*
10	7	The track is within 160 km of a sensitive point.
8	5	The track is between 160 km and 190 km of a sensitive point.
6	3	The track is further than 190 km from a sensitive point.

\*Missile ranges for various weapons are given at <http://www.fas.org/man/dod-101/sys/smart/index.html>. The HPM bases the evaluation criteria for (X, Y) position on known blue missile ranges compounded with air speed of known blue aircraft. Most common missiles aboard invading aircraft cannot strike outside a range of 70 km.

**Table 2-5. Air Speed evaluation criteria.**

Rating (points)	Evaluation Criteria---Item 2: Air Speed
10	The track is traveling at speeds over 400 m/s.
9	The track is traveling at speeds between 250 m/s and 400 m/s.
8	The track is traveling at speeds between 150 m/s and 250 m/s.
7	The track is traveling at speeds below 150 m/s.

**Table 2-6. Altitude evaluation criteria.**

Rating (points)	Evaluation Criteria---Item 3: Altitude
4	The track is above 12 km.
6	The track is between 6 and 12 km.
8	The track is between 3.5 and 6 km.
10	The track is below 3.5 km.

**Table 2-7. Heading evaluation criteria.**

Rating (points)	Evaluation Criteria---Item 4: Heading
10	The track is within 20 degrees of heading towards an asset.
6	The track is between 20 and 40 degrees of heading towards an asset.
4	The track is between 40 and 70 degrees of heading towards an asset.
0	The track is more than 70 degrees of heading towards an asset.

**Table 2-8. Track Type evaluation criteria.**

Rating (points)	Evaluation Criteria---Item 5: Track Type
10	The track is a low observable.
8	The track is a fighter.
8	The track is a bomber.
8	The track is an airborne commander.
5	The track is an unmanned aerial vehicle (UAV).

### **2.1.3 Track Prioritization**

Track prioritization, task 7 in the SOC operator model in Figure 2-2, arranges the threat evaluated tracks from the previous task into order of lethality for the track assignment. The SOC can use this prioritization to focus its reaction on the highest priority (i.e., most dangerous) tracks in the air sector. Ignoring a highly threatening track until other lesser threatening tracks have been analyzed could be detrimental to the IADS.

### **2.1.4 Track Assignment**

The SOC will assign tracks for engagement, task 8 in the SOC operator model in Figure 2-2, if they pose a serious threat to the IADS. That assignment would include any tracks with a threat score above a predetermined cutoff level. The track assignment cutoff level is a notional number set by the user. No assignment occurs for tracks having a threat score below the cutoff since the SOC considers these tracks unthreatening. Additionally, if a track has been assigned because it

once was deemed threatening (i.e. above the cutoff level) but then becomes unthreatening in a subsequent track update (i.e. below the cutoff level) the track assignment for that track will be cancelled.

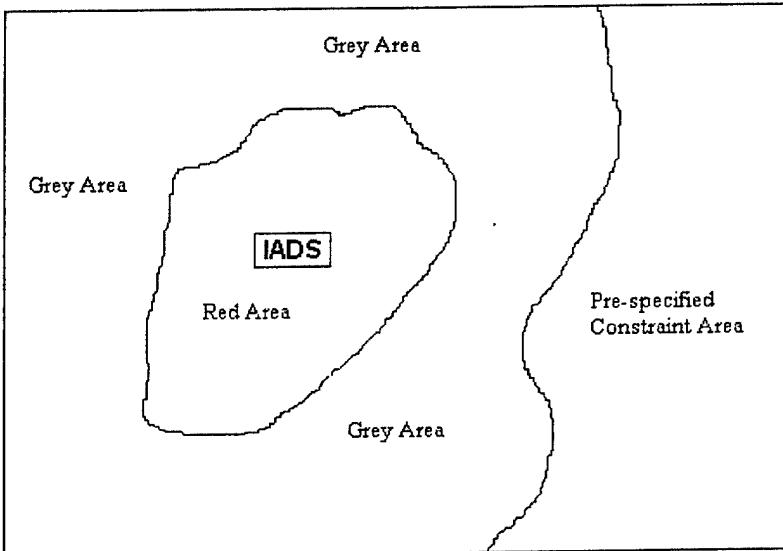
Human behavioral variations caused by cultural factors can be made by altering the track assignment cutoff level. Relatively speaking, decreasing the cutoff will cause the SOC to assign tracks at a lower threat level, which in turn can cause more track assignments to be made and earlier assignment times. In terms of track cancellations, decreasing the cutoff will cause the SOC to cancel tracks less frequently with associated later cancellation times. Increasing the cutoff will have the opposite effect for both assignments and cancellations.

The track assignment process has a different algorithm for each of three war scenarios: prewar, traditional, and SAMbush. The following pages discuss each algorithm, named after the scenarios, in detail.

#### *Prewar*

The prewar scenario finds the IADS in a precautionary posture as two enemy airplanes fly a nearby border patrol mission. As the enemy airplanes maneuver near the IADS they enter and exit air zones triggering various IADS behaviors. This scenario has three air spaces as shown in Figure 2-3.

The first air space (far right) is the pre-specified constraint area (PCA). The SOC does not permit the SAM Ops to release weapons within this space. Reasons for restricting weapons release in this space range from infringing upon other countries airspaces to internal air zones where a country can safely fly its own aircraft without the fear of being shot (i.e. prevention of fratricide).



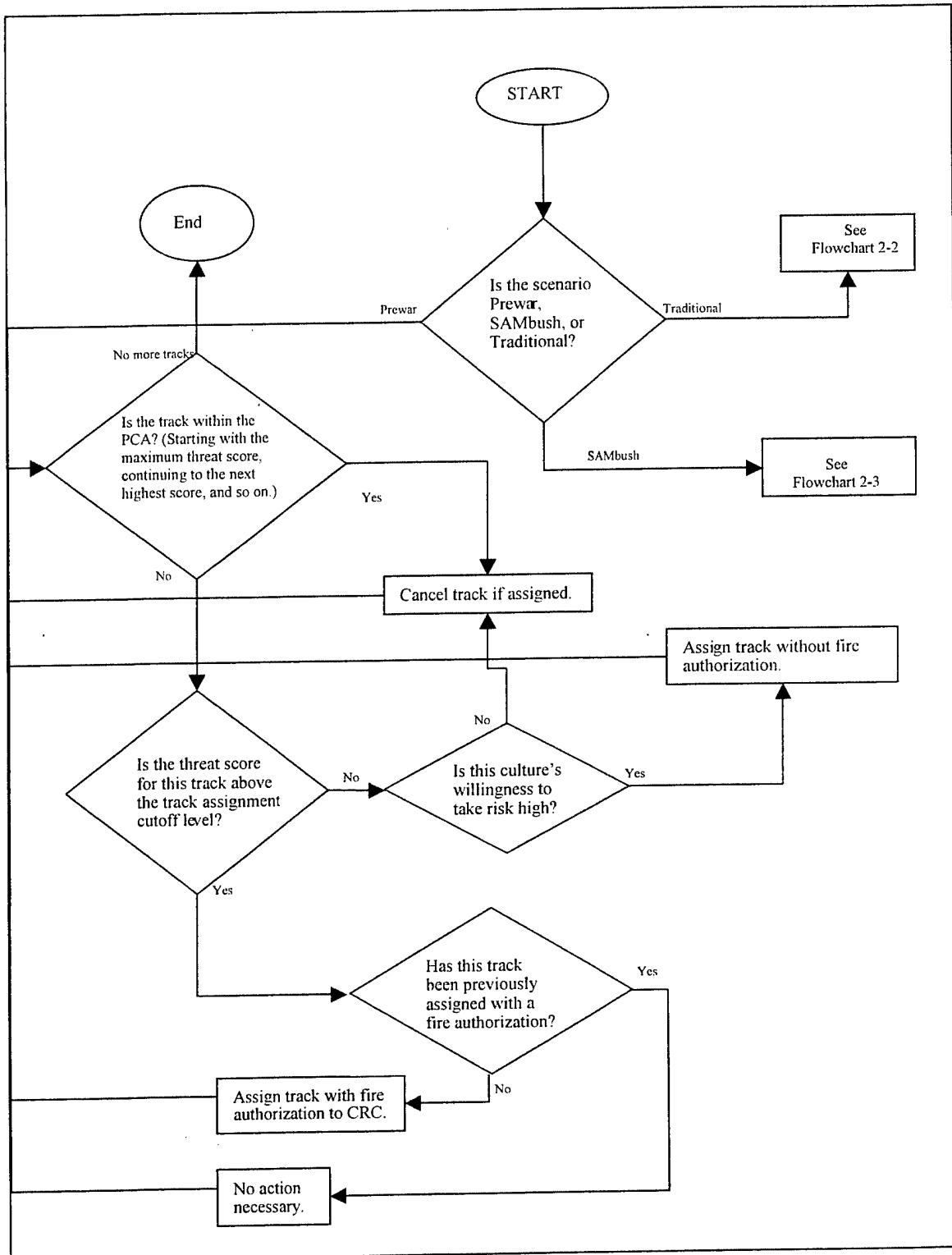
**Figure 2-3. IADS Air Zones.**

The space formed within the cutoff level that envelops the IADS is called the red area. Any enemy aircraft detected by the IADS within this area are typically engaged and fired upon.

The third space, called the grey area, is the area outside the red area and not in the PCA. It is in this area that behavior of the IADS operations may differ. Aircraft within this area are not yet threatening enough to require weapons release, but can easily, within a few minutes, become so. Operator and command and control differences can significantly alter how aircraft within the gray area are handled. In some cultures, acquisition radar will "light up" aircraft within the grey area, having no intention of weapons release, to frighten the enemy aircraft into retreating or

alternatively to provoke the enemy aircraft into attacking and providing the IADS with an opportunity to down enemy aircraft. Other cultures will not risk acquiring any aircraft in this grey area. MA&D modeled the behavioral difference between cultures for aircraft within the grey area using the Willingness to take Risk (WR) cultural parameter. North Korea, with a high WR, will assign tracks within the grey area for acquisition (“light up”), but not weapons release. Iraq, with a low WR, will not assign tracks within the grey area. MA&D scripted the prewar scenario to contrast the possible different track assignment behaviors.

Flowchart 2-1 reveals the algorithm modeled for prewar track assignment. To acquire and “light up” enemy aircraft the SOC must give track assignments without fire authorization. For full weapons release authority, the SOC must give track assignments with fire authorization.

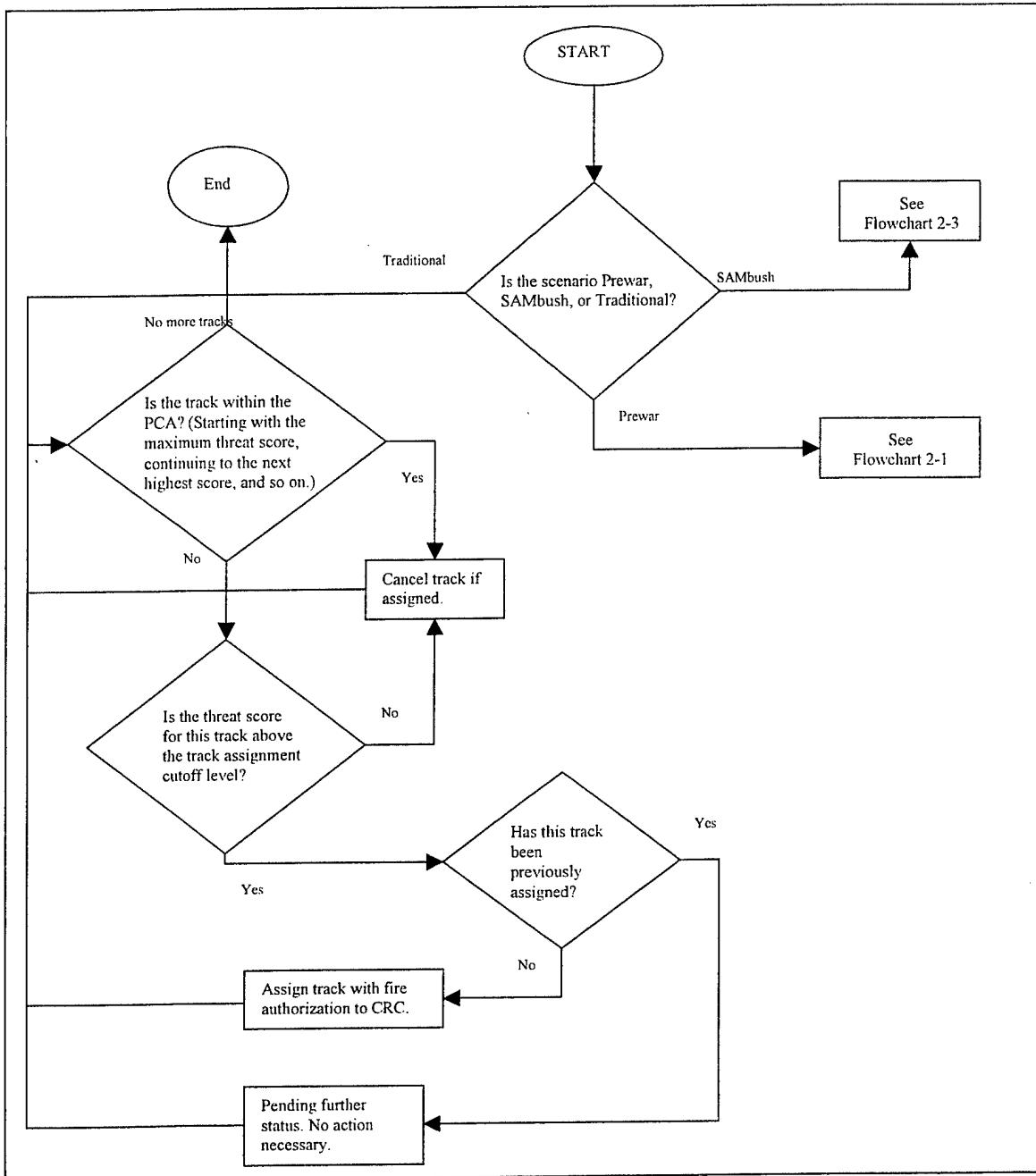


Flowchart 2-1. Prewar Track Assignment.

*Traditional*

The traditional scenario finds the IADS in a (full-blown) wartime situation with a defensive posture. As enemy aircraft enter the IADS air space, the SOC will only assign aircraft with threat scores above the cutoff level, always using track assignments with fire authorization. Since this scenario is in a full blown war scenario, the IADS will not waste time by assigning tracks in the grey area and therefore will not use track assignments without fire authorization. Cultural behavioral differences in the traditional scenario are seen by the distances and times at which the SOC assigns and cancels tracks, as mentioned in sub-section 2.1.2, Threat Evaluation.

Flowchart 2-2 reveals the algorithm modeled for traditional track assignment. In this scenario, the SOC orders only track assignments with fire authorization.



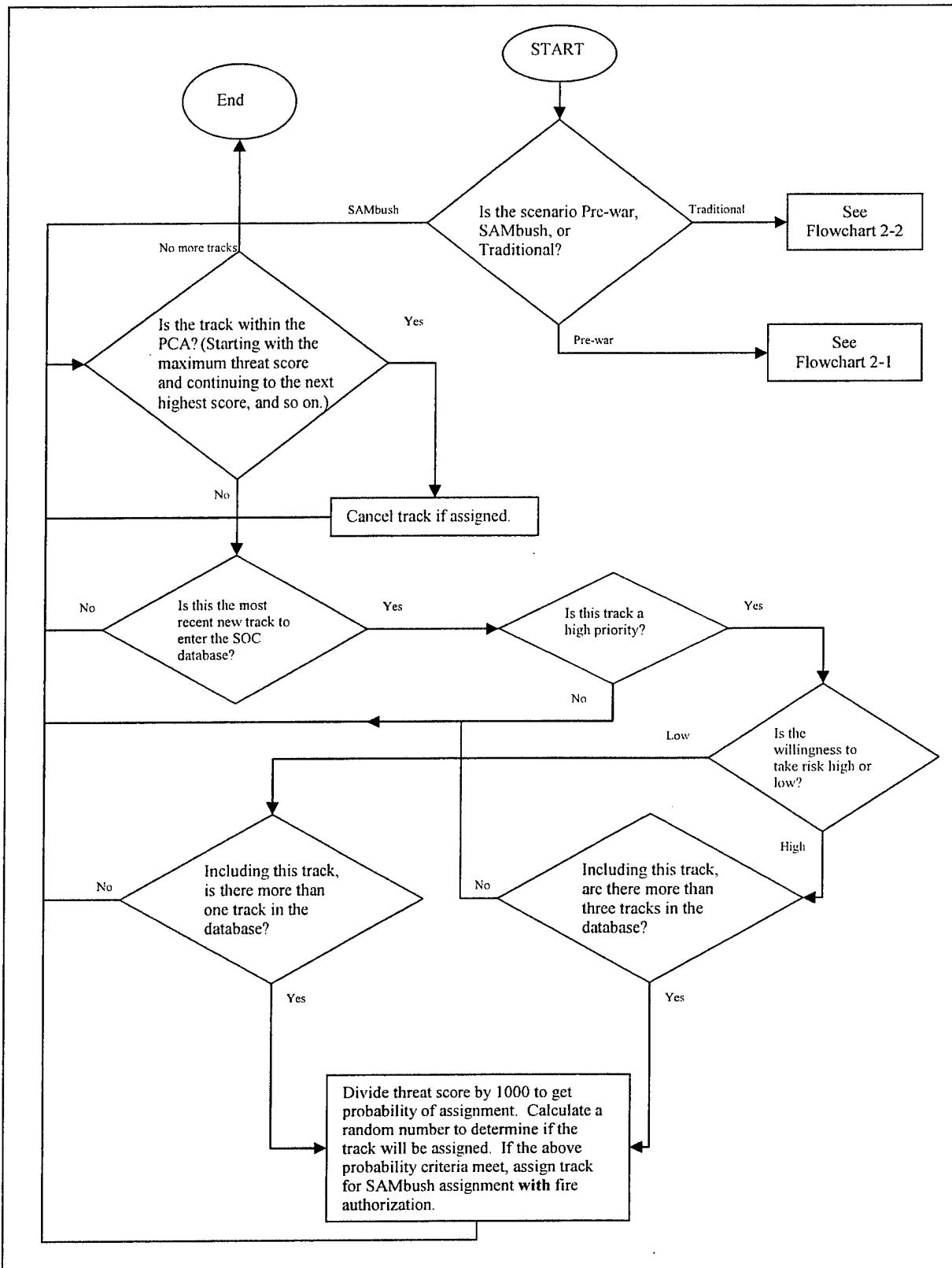
Flowchart 2-2. Traditional Track Assignment.

### *SAMBUSH*

The SAMbush (SAM ambush) scenario finds the IADS in an unconventional wartime situation with the sole intention of using unexpected attacks to catch the enemy off guard. Enemy aircraft flying in the air sector believe that the IADS has been compromised and get a false sense of security. The IADS will use this situation to its advantage and allow aircraft to enter the air sector, furthering an increase in the enemy aircraft's comfort level, whereupon the SOC will order a sudden SAMbush track assignment with fire authorization. A SAMbush track assignment with fire authorization is very similar to a track assignment with fire authorization, however it gives further direction to the SAM Ops to acquire an enemy and release a SAM quickly to produce an element of surprise. If the SAM Ops has the acquisition radar on too long, it reveals SAM Ops position information to the enemy, thereby negating further SAMbush leverage from that location and gives increased opportunity for the aircraft to escape unharmed.

Cultural behavioral variations in the SAMbush scenario are modeled based on the number of aircraft the IADS will allow into its air sector before the SOC gives a SAMbush track assignment; North Korea allows more flyovers than Iraq. MA&D designed the algorithm so that the SOC gives SAMbush track assignments only to high priority tracks (i.e., fighter, bomber, or stealth airplanes). MA&D came to this conclusion because it felt that an IADS would not waste a SAMbush opportunity on aircraft of lesser value, such as a UAV. MA&D used the WR cultural parameter to determine how many aircraft each culture would allow in the air sector before a SAMbush attack. Since, by our definition, a high WR indicates that an individual will make decisions that place him in vulnerable situations, IADS having a high WR will wait until more aircraft enter its air sector than cultures with a low WR. MA&D set the number of aircraft that a culture with a high WR will let enter its air sector before a SAMbush assignment to four (4), and a culture with a low WR to two (2). The SAMbush algorithm also includes a bit of probability to determine if the SOC orders a SAMbush assignment by generating a random number between zero and 1000. If the random number is less than or equal to the track threat score, the SOC will finally give the track a SAMbush assignment. Adding this logic into the SAMbush algorithm gives a more realistic assignment behavior, where tracks with higher threat score levels, have a greater possibility of getting assigned for a SAMbush.

Flowchart 2-3 reveals the algorithm modeled for SAMbush track assignment. In this scenario the SOC gives only SAMbush track assignments with fire authorization.

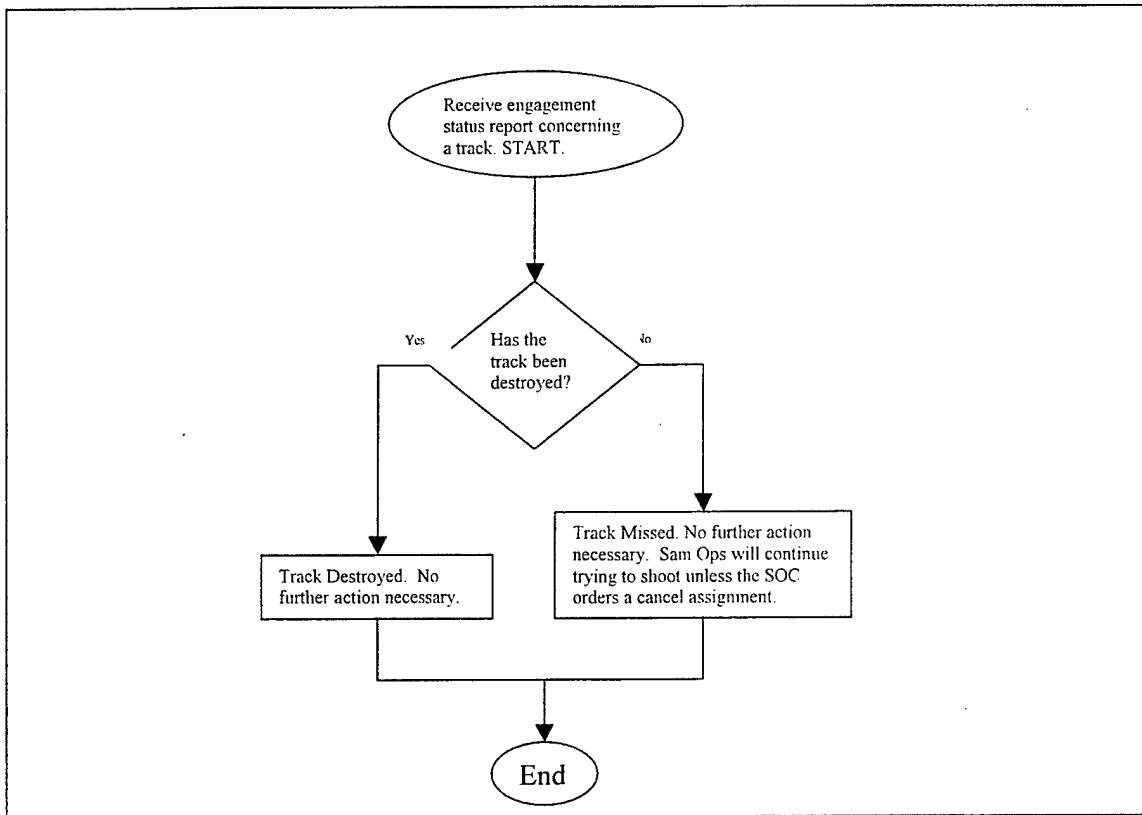


Flowchart 2-3. SAMbush Track Assignment.

### 2.1.5 Receive Engagement Status Reports

The SOC receives engagement status reports describing the state of an engagement after a SAM has been released. There are two variations of this report; either the track was hit and destroyed, or the track was missed and is still at large.

When the SOC receives an engagement status report concerning a track, it will take the following actions as described by Flowchart 2-4. If it has been destroyed then no further action is necessary. If the track has not been destroyed, the SAM Ops will continue trying to shoot until the SOC orders a cancel assignment order.



Flowchart 2-4. Receive Engagement Status Report.

### 2.2 Control and Reporting Center (CRC)

The CRC is the C2 node in the IADS that is one echelon below the SOC and one above the CRP. Its purpose is to serve as a mediator between the SOC and all CRPs by propagating track information from CRPs to the SOC, delivering engagement status reports from SAM Ops to the SOC, and disseminating SOC track assignment and cancellation orders to the SAM Ops. Figure 2-4 shows the task network model of the command and control functionality of the CRC.

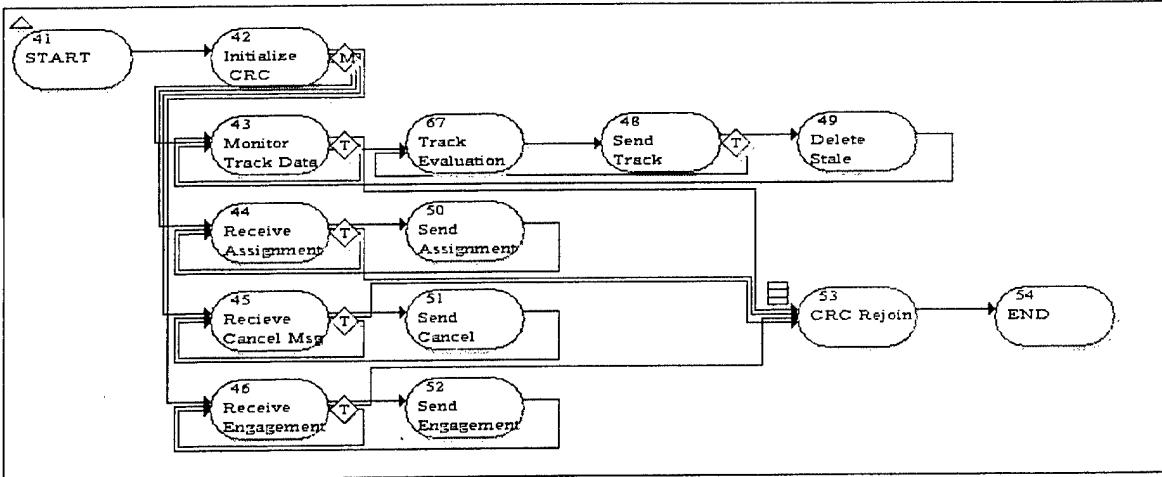


Figure 2-4. CRC task network.

This HPM abstracts the complex behaviors performed by the CRC into ten functions: 1) Track Data Monitoring, 2) Track Evaluation, 3) Sending Track Data, 4) Deleting Stale Track Data, 5) Receiving Track Assignment Messages, 6) Sending Track Assignment Messages, 7) Receiving Track Cancel Messages, 8) Sending Track Cancel Messages, 9) Receiving Engagement Status Reports, and 10) Sending Engagement Status Reports.

When JIMM has subscribed for CRC HPM services, task 41, "Start," commences leading to four task paths: Task 42, "Initialize CRC," Task 43, "Receive Assignment Messages," Task 45, "Receive Cancel Messages," and Task 46, "Receive Engagement Status Reports." Each path leads to task sections that the CRC will perform concurrently until the simulation finishes. The first section is the set of tasks from task 43, "Monitor Track Data," to task 49, "Delete Stale Tracks." Of the human processes modeled at the CRC, this set is the most complex and is discussed in more detail in the next sub-sections. The remaining three sections are two task processes that receive information in the first task and send out the same information in the second task: tasks 44 and 50 handle SOC track assignment messages, tasks 45 and 51 handle SOC track cancellation messages, and tasks 46 and 52 handle engagement status reports. Each of these two task processes simulates human behaviors found at a CRC: waiting for communication, processing communication, and sending communication. The primary task, in each case, replicates waiting for a new message. After receipt, the second task processes the communication (i.e. recognizes the form and content of information) and then repeats the communication to the appropriate destination. All engagement status reports travel upward to the SOC from the CRC, exactly like track data messages. SOC track assignment and cancellation messages go to the CRC within JIMM and are handled there internally, later to be sent to the appropriate SAM Ops in JIMM.

### 2.2.1 Monitoring Track Data

The CRC must continuously monitor incoming track data from all CRPs to send up to the SOC. Two and three dimensional radar devices supply track data of the air sector to the CRC via the CRP with detailed information regarding detected aircraft. At the moment the CRC receives new track data, it must perform track evaluation (discussed in the next sub-section) to determine what track data to send up to the SOC unless the CRC has already received track data and is in the middle of evaluation (in task 67, "Track Evaluation," task 48, "Send Track Data," and task 49,

“Delete Stale Track Data.”). Upon returning to “Monitoring Track Data” (task 43), the CRC will confront the new data by immediately advancing into “Track Evaluation” (task 67).

The model replicates this human operator action of waiting, or monitoring track data, by continuously performing the “Monitor Track Data” of task 43. When the CRC receives new track data the looping ceases and task flow continues to “Track Evaluation” (task 67).

### 2.2.2 *Track Evaluation*

Track evaluation (task 67) involves the CRC deciphering what track data to pass up to the SOC. Most all track data is passed along to the SOC except when multiple instances of the same track have been received by the CRC. A possibility exists that more than one track on an aircraft will surface as a result of having overlapping radar sensors, a typical feature of most IADS. When this redundancy occurs, the CRC will evaluate the perception times included with the track data and choose the most recent track data message to send to the SOC.

### 2.2.3 *Send Track Data*

After completing track evaluation, the CRC will send the chosen track data message to the JIMM CRC, where it will then be repeated to the CART SOC. The CRC only evaluates and sends one track at a time. If more tracks for different air threats were received (in task 43), then the CRC will (loop back to task 67 to) perform track evaluation on the next track in the receiving queue. After all track data messages have been sent to the JIMM CRC in the receiving queue, path flow continues (to task 49).

### 2.2.4 *Delete Stale Track Data*

Deleting stale track data is an important step in the maintenance of track data. To avoid wasting time worrying about track data that is obsolete, or “stale,” the CRC invokes a function to delete stale track data from its database. Criteria differ, but generally an IADS will drop a track after a certain amount of missed updates. If expecting an update on a track once every period of a radar sweep, one can set the function to delete stale tracks in multiples of the radar sweep time.

## 2.3 Control and Reporting Post (CRP)

The CRP is the C2 node in the IADS that is one echelon below the CRC and one echelon above the radar sensor systems. It receives track data from radar sensor systems and sends this information up to the CRC. Figure 2-5 shows the task network model of the command and control functionality of the CRP.

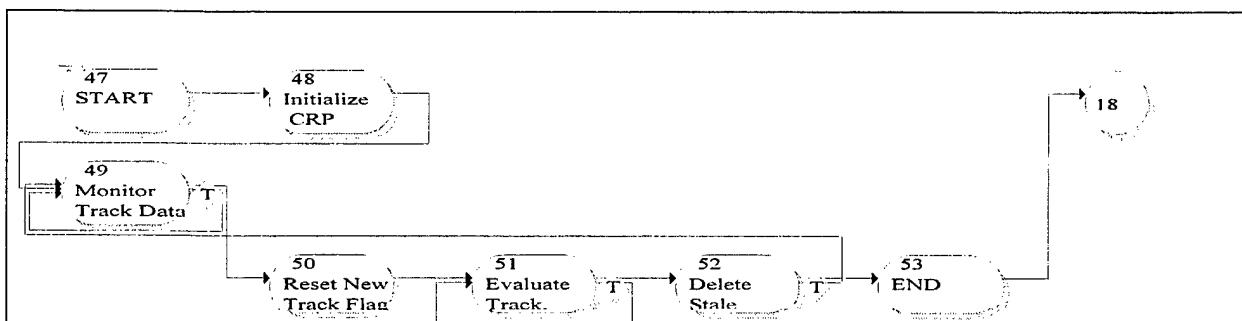


Figure 2-5. CRP task network.

When a JIMM entity subscribes for CRP HPM services, task 47, the “Start” task, commences. This “Start” task initiates the tasks required of a CRP for each track encountered. It starts a task path to task 48, “Initialize CRP.” This task initializes the track evaluation process that will be discussed in more detail in sub-section 2.3.2 and then starts a repeating set of tasks starting with task 49, “Monitor Track Data” and continuing to task 52, “Delete Stale Tracks.”

### **2.3.1 *Monitoring Track Data***

The CRP must continuously monitor incoming track data to send up to the CRC. Two and three dimensional radar devices supply track data of the air sector to the CRP, with detailed information regarding detected aircraft. Track data must be evaluated (discussed in the next sub-section) to determine what track data to send up to the CRC unless the CRP has already received track data and is in the middle of evaluation (task 51, “Track Evaluation,” and task 52, “Delete Stale Track Data”). Upon returning to “Monitoring Track Data” (task 49), the CRP will confront the new data by immediately advancing into “Track Evaluation” (task 51).

The model replicates this human operator action of waiting, or monitoring track data, by continuously performing the “Monitor Track Data” of task 49. When the CRC receives new track data the looping ceases and task flow continues to “Track Evaluation” (task 51).

### **2.3.2 *Track Evaluation***

Track evaluation (task 51) involves the CRP deciphering what track data to pass up to the CRC. Most all track data is passed along except when multiple instances of the same track have been received. A possibility exists that more than one track on an aircraft will surface as a result of having overlapping radar sensors, a typical feature of most IADS. When this redundancy occurs, the CRP will evaluate the perception times included with the track data and choose the most recent track data message to send to the CRC.

### **2.3.3 *Delete Stale Track Data***

Deleting stale track data is an important step in the maintenance of track data. To avoid wasting time worrying about track data that is obsolete, or “stale,” the CRP invokes a function to delete stale track data from its database. Criteria differ, but generally an IADS will drop a track after a certain amount of missed updates. If expecting an update on a track once every period of a radar sweep, one can set the function to delete stale tracks in multiples of the radar sweep time.

## **3.0 *Data Messages***

Six message types are communicated between JIMM and CART: 1) Initialization Data, 2) Track Data, 3) Track Assignment Data, 4) Track Cancellation Data, 5) Engagement Status Data, and 6) Height Finding Request Data (see Table 3-1 through Table 3-6). This section describes the messages, giving the data types contained in each message and its definitions. For more details concerning these messages, please refer to the Interface Control Document (ICD) included with the Cultural Modeling of Command and Control (CMC2) Final Report.

**Table 3-1. Initialization data.**

Data Type	Description
Entity Identification Number	Unique number that identifies this entity from other subscribing entities.
Entity Type	Indicates the type of human performance modeling node JIMM requests for the subscribing entity (i.e. SOC, CRC, or CRP).
Entity X Position	X coordinate of the entity from scenario center (meters).
Entity Y Position	Y coordinate of the entity from scenario center (meters).
Entity Familiarity with the Enemy Level	Cultural parameter indicating how familiar the entity is with the enemy (i.e. High or Low).
Entity Willingness to take Risks Level	Cultural parameter indicating how willing the entity is to take on risk (i.e. High or Low).
Entity Distribution of Power Level	Cultural parameter indicating how authority is distributed amongst the military (i.e. High or Low).
Number of Subordinates	Number of subordinates beneath the subscribing entity.
Subordinate Entity Identification Numbers	Identification numbers of subordinate entities.
Subordinate Entity Types	Subordinate entity types.
Subordinate Entity X Positions	X coordinate of the subordinates from scenario center (meters).
Subordinate Entity Y Positions	Y coordinate of the subordinates from scenario center (meters).
Scenario Type	War scenario type for the subscribing entity (i.e. prewar, traditional, or “SAMbush”).
Mode of Operations	Operations mode for the subscribing entity (i.e. manual or automatic).

**Table 3-2. Track Data Message.**

Data Type	Description
Sender Entity Identification Number	Identifies the specific sender of the track data.
Track Entity Identification Number	Unique number given to the track originated by the radar.
Receiver Entity Identification Number	Unique number given to identify the recipient of the track data message.
Track X position	X coordinate of the track from scenario center (meters).
Track Y position	Y coordinate of the track from scenario center (meters).
Track Heading	Heading of the track relative to true north (degrees), clockwise is positive.
Track Speed	Speed of the track (meters per second).
Track Altitude	Z coordinate of the track (meters).
Track Type	Classification of the aircraft (i.e. Bomber, Fighter, Low Observable, Airborne Command, Other or UAV).
IFF Enumeration	Identification of Friend or Foe (i.e. Friend, Hostile, Civilian, Unknown).

**Table 3-3. Track Assignment Data Message.**

Data Type	Description
Sender Entity Identification Number	Identifies the specific sender of the track data.
Receiver Entity Identification Number	Unique number given to identify the recipient of the track data message.
Track X position	X coordinate of the track from scenario center (meters).
Track Y position	Y coordinate of the track from scenario center (meters).
Track Heading	Heading of the track relative to true north (degrees), clockwise is positive.
Track Speed	Speed of the track (meters per second).
Track Altitude	Z coordinate of the track (meters).
Track Type	Classification of the aircraft (i.e. Bomber, Fighter, Low Observable, Airborne Command, Other or UAV).
IFF Enumeration	Identification of Friend or Foe (i.e. Friend, Hostile, Civilian, Unknown).
Fire Authorization Type	Identifies the fire authorization for the track assignment (i.e. track assignment without fire authorization, track assignment with fire authorization, SAMbush assignment with fire authorization).
Track Identification Number	Unique number to identify which track the SOC has assigned.

**Table 3-4. Track Cancellation Data Message.**

Data Type	Description
Sender Entity Identification Number	Identifies the specific sender of the track data.
Receiver Entity Identification Number	Unique number given to identify the recipient of the track data message.
Track Identification Number	Unique number to identify which track the SOC has assigned.
Cancellation Type	Identifies the type of cancellation (i.e. destroy missile in flight, or let fly.) The HPM has been designed to always “let fly” if the cancellation has arrived after weapons release. If the cancellation arrives before weapons release, the missile will not fly at all.

**Table 3-5. Engagement Status Report Message.**

Data Type	Description
Sender Entity Identification Number	Identifies the specific sender of the track data.
Track Identification Number	Unique number to identify which track the SOC has assigned.
Engagement Status	Identifies the engagement status (i.e. fired-on and destroyed or fired-on and missed).

**Table 3-6. Height Finding Request Data.**

Data Type	Description
Sender Entity Identification Number	Identifies the specific sender of the track data.
Receiver Entity Identification Number	Unique number to identify the recipient.
Track Identification Number	Unique number to identify the track.

## 4.0 Developing Limitations

Due to developing limitations in JIMM, the actual flow of SOC track assignment and cancellation messages goes only from the CART SOC to the JIMM SOC. For similar reasons, engagement status reports only travel from the JIMM CRC to the CART SOC. Nonetheless, MA&D developed the CART IADS HPM to handle the message flow described above. Therefore, with no modification necessary, the CART HPM will be able to provide these services in the future when JIMM becomes capable.